

UNIVERSITÉ DE MONTRÉAL

SIMULATEUR COLLABORATIF DE CHIRURGIE D'INSTRUMENTATION  
DU RACHIS SCOLIOTIQUE EN RÉALITÉ VIRTUELLE :  
ÉTUDE DE FAISABILITÉ

JACQUES-ANDRÉ BOULAY  
DÉPARTEMENT DE GÉNIE INFORMATIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE INFORMATIQUE)  
DÉCEMBRE 2006



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file    Votre référence*

*ISBN: 978-0-494-25534-6*

*Our file    Notre référence*

*ISBN: 978-0-494-25534-6*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

SIMULATEUR COLLABORATIF DE CHIRURGIE D'INSTRUMENTATION  
DU RACHIS SCOLIOTIQUE EN RÉALITÉ VIRTUELLE :  
ÉTUDE DE FAISABILITÉ

présenté par : BOULAY Jacques-André

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

Mme. CHERIET Farida, Ph.D., présidente

M. OZELL Benoît, Ph.D., membre et directeur de recherche

M. AUBIN Carl-Éric, Ph.D., membre et codirecteur de recherche

M. LABELLE Hubert, Ph.D., M.D., membre externe

*À mes parents, mon frère, ma tante, Luc,  
Pierre, Olivier et M. K. pour leur  
présence, support et encouragement.*

*À moi comme preuve d'une évolution,  
d'un aboutissement et d'une  
conquête face à l'adversité.*

## Remerciements

Je tiens à remercier mes directeurs de recherche, Benoît Ozell et Carl-Éric Aubin, pour leurs conseils judicieux, leurs encouragements et le guidage qu'ils m'ont offert durant cette période. Je me sens sincèrement privilégié d'avoir eu la chance de travailler avec eux. Ils sont pour moi des modèles de réussite à suivre en tant que chercheurs. L'opportunité passionnante qu'ils m'ont offerte risque de modifier à jamais le cours de ma vie. Je ne les remercierai jamais assez.

Je dois également adresser un remerciement supplémentaire à Carl-Éric Aubin pour l'incalculable privilège de son aide financière au travers de ses fonds CRSNG. Alexandre Boivin et Nathalie Bourassa, du Laboratoire de Modélisation Biomécanique et de Chirurgie Assistée par Ordinateur (LMBCAO), doivent être remerciés pour leur aide précieuse concernant l'utilisation du logiciel S3. Je remercie également les chercheurs, assistants de recherche et étudiants de ce laboratoire pour leur accueil, leur aide générale et le temps qu'ils ont investi afin de faciliter mon adaptation.

Les spécialistes en orthopédie, Stefan Parent et Jean-Marc Mac-Thiong, doivent également être remerciés pour leur participation aux tests usagers.

Je remercie finalement Marie-Ève Plamondon, de la Direction Enseignement et Formation de l'École Polytechnique de Montréal, pour les feuilles de style qu'elle m'a procurées pour faciliter le processus de rédaction de ce mémoire.

## Résumé

Les patients atteints d'une scoliose idiopathique sévère doivent subir une intervention d'instrumentation chirurgicale hautement invasive et risquée. Pourtant, il n'existe aucun outil informatique d'entraînement collaboratif qui simule fidèlement l'expérience d'une telle chirurgie et qui permet une planification préopératoire pour un patient donné. L'objectif de cette recherche est d'évaluer la faisabilité du développement d'un tel simulateur à l'aide de dispositifs de réalité virtuelle (CAVE et Immersadesk), d'une modélisation biomécanique de la colonne vertébrale et de fonctionnalités collaboratives. Elle vise également à déterminer si un tel outil pourrait être utilisé efficacement pour l'entraînement des chirurgiens orthopédistes lors de sessions de télé-enseignement.

Ces objectifs ont motivé la création d'un prototype, le *Collaborative Virtual Reality Spine Surgery Simulator* (CVrS3), simulant une salle de chirurgie virtuelle où plusieurs chirurgiens, géographiquement éloignés les uns des autres, peuvent mutuellement se voir et collaborer lors de sessions d'entraînement à la chirurgie d'instrumentation du rachis scoliotique. Le système développé est constitué d'un serveur biomécanique collaboratif, d'un serveur de téléprésence multi-usagers et d'un client de simulation de chirurgie scoliotique en réalité virtuelle.

Le serveur biomécanique intègre une partie du code source du logiciel *Spine Surgery Simulator* (S3) développé au Laboratoire de Modélisation Biomécanique et de Chirurgie Assistée par Ordinateur (LMBCAO). Ce laboratoire est partagé par l'École Polytechnique de Montréal et le Centre de Recherche du CHU Sainte-Justine. S3 est un simulateur de chirurgie scoliotique pour ordinateur conventionnel permettant de tester rapidement, avant une réelle chirurgie, différentes stratégies d'intervention pour un patient donné. Il modélise le comportement du rachis scoliotique lors de l'instrumentation. Le serveur développé adapte cette modélisation pour une utilisation collaborative en réalité virtuelle. Les clients s'y connectent pour accéder, d'une manière concurrente, à l'état du rachis du patient.

Le rôle du serveur de téléprésence multi-usagers est de propager les déplacements de chaque participant à tous les autres. Les clients utilisent ces informations pour gérer le déplacement des avatars, associés à chaque participant, à l'intérieur de la salle de chirurgie virtuelle. Une meilleure perception de la présence et des intentions des participants est ainsi possible.

Le client de simulation est une application infographique tridimensionnelle qui affiche et gère une salle de chirurgie virtuelle interactive. Il permet de se déplacer à l'intérieur de la salle, d'interagir avec les objets en faisant partie (lampe, implants, outils, etc.) et d'effectuer des manoeuvres chirurgicales sur le rachis scoliotique d'un patient spécifique. Deux versions ont été développées : immersive et non-immersive. La version immersive est conçue pour les systèmes de réalité virtuelle avancés tels que la CAVE et l'Immersadesk. La version non-immersive procure plus de flexibilité en offrant une interface sur ordinateur conventionnel avec clavier et souris.

Le présent mémoire présente les choix architecturaux, les caractéristiques et les particularités associés aux entités développées. Les tests usagers, effectués avec deux spécialistes en orthopédie et trois spécialistes en environnements virtuels, sont présentés. Ces tests ont permis d'évaluer les lacunes et les bénéfices du prototype développé. Ils ont également permis d'évaluer l'utilité potentielle de ce type de simulateur en contexte clinique. Les résultats de ces tests sont présentés et commentés.

Ce travail a prouvé la faisabilité du développement d'un simulateur collaboratif de chirurgie scoliotique en réalité virtuelle intégrant la plus récente version du module S3 de modélisation biomécanique du rachis scoliotique. Il a également prouvé la supériorité des systèmes immersifs de réalité virtuelle par rapport aux interfaces conventionnelles, avec écran, clavier et souris standard, en ce qui a trait au réalisme de la simulation et à la facilité d'interaction. Les tests usagers effectués, bien que de petite envergure, ont mis en évidence l'utilité que pourrait avoir ce type de simulateur pour la formation générale des médecins résidents, la compréhension de la biomécanique du rachis scoliotique, le

télé-enseignement et le télé-entraînement à la chirurgie d'instrumentation du rachis scoliotique.



## **Abstract**

Patients suffering from a severe idiopathic scoliosis are subject to a highly invasive and risked instrumentation surgery. However, there is no collaborative training computer tool which simulates truthfully the experience of such a surgery and that allows a preoperative planning for a given patient. The objective of this research is to assess the feasibility of the development of such a simulator with the help of virtual reality technologies (CAVE and Immersadesk), a biomechanical modelling of the spinal column and collaborative functionalities. It also aims to determine if such a tool could be efficiently used for orthopaedist surgeons training during distance learning sessions.

These objectives justified the creation of a prototype, the Collaborative Virtual Reality Spine Surgery Simulator (CVrS3), which simulates a virtual surgery room where several surgeons, geographically away, can mutually see each other and collaborate during scoliosis spine surgery training sessions. The developed system is composed of a biomechanical collaborative server, a telepresence multi-users server and a virtual reality scoliosis spine surgery simulation client.

The biomechanical server uses the source code from the Spine Surgery Simulator (S3) software developed in the Laboratory of Biomechanical Modeling and Computer-Aided Surgery. This laboratory is shared by Ecole Polytechnique de Montreal and the Research Center of Sainte-Justine University Hospital Center. S3 is a spine surgery simulator for conventional computer allowing to test rapidly, before a real surgery, different intervention strategies for a given patient. S3 models the behavior of the scoliotic spine during instrumentation. The developed server adapts this modeling for a collaborative virtual reality use.

The telepresence multi-users server role is to propagate every participant movement to all the others. It permits the clients to move the user associated avatars inside the virtual surgery room. A better perception of participant's presence and intentions is then possible.

The simulation client is a three-dimensional graphic application that displays and manages an interactive virtual surgery room. It allows to move inside the room, to interact with objects (lamp, implants, tools, etc) and to perform surgical maneuvers on the scoliotic spine of a specific patient. Two versions were developed: immersive and non-immersive. The immersive version is made for advanced virtual reality systems such as the CAVE and the Immersadesk. The non-immersive version brings more flexibility by providing a conventional computer interface with keyboard and mouse.

This thesis introduces the architectural choices and features associated with the developed entities. A description of the user testing, performed with two orthopaedics specialists and three virtual environments specialists, is also provided. The tests allowed to assess drawbacks and benefits of the developed prototype. They also allowed to evaluate the potential usefulness of this type of simulator. The results from the tests are presented and commented.

This research proved the feasibility of the development of a virtual reality collaborative spine surgery simulator based on the most recent version of S3 biomechanical modeling. It also proved the superiority of virtual reality immersive systems against conventional interfaces based on simulation realism and easiness of interaction. The user testing, although of small scale, points out this kind of simulator could really be useful for resident general training, the understanding of the spine biomechanics, distance learning and training of the scoliosis spine surgery.

## Table des matières

DÉDICACE .....	iv
REMERCIEMENTS .....	v
RÉSUMÉ .....	vi
ABSTRACT .....	ix
TABLE DES MATIÈRES .....	xi
LISTE DES ANNEXES .....	xvi
LISTE DES TABLEAUX .....	xvii
LISTE DES FIGURES .....	xviii
LISTE DES SIGLES ET ABRÉVIATIONS .....	xxi
 <b>CHAPITRE 1 : INTRODUCTION .....</b>	<b>1</b>
1.1 Motivation de la recherche .....	1
1.1.1 Définition de la scoliose idiopathique .....	2
1.1.2 Traitements de la scoliose .....	3
1.1.3 Outils chirurgicaux .....	5
1.2 Contexte de la recherche .....	10
1.3 Éléments de problématique .....	13
1.4 Objectifs de recherche .....	15
1.5 Contributions et originalité des travaux .....	16
1.6 Organisation du mémoire .....	17
 <b>CHAPITRE 2 : REVUE DE LA LITTÉRATURE .....</b>	<b>18</b>
2.1 Outils et simulateurs médicaux en réalité virtuelle .....	18
2.1.1 La réalité virtuelle : définition et caractéristiques .....	18
2.1.2 Domaines d'application de la réalité virtuelle .....	21
2.1.3 Les outils et simulateurs médicaux en réalité virtuelle .....	22
2.2 Environnements virtuels de collaboration .....	28

2.2.1	Les EVC : définition et caractéristiques .....	28
2.2.2	Domaines d'application des EVC .....	32
2.2.3	Les applications collaboratives et EVC médicaux .....	36

### **CHAPITRE 3 : ÉLABORATION DE L'ARCHITECTURE DU SIMULATEUR .. 43**

3.1	Architecture proposée.....	43
3.1.1	Serveur biomécanique d'instrumentation collaborative du rachis scoliotique .....	45
3.1.2	Serveur de téléprésence multi-usagers .....	46
3.1.3	Client de simulation de chirurgie scoliotique en réalité virtuelle...	47
3.2	Outils et données initiales.....	51
3.2.1	CAVE et Immersadesc .....	51
3.2.2	VR Juggler et OpenSG.....	52
3.2.3	C++ Sockets Library .....	53
3.2.4	Spine Surgery Simulator.....	53
3.2.5	Virtual Spine Surgery Simulator.....	54

### **CHAPITRE 4 : SERVEUR BIOMÉCANIQUE D'INSTRUMENTATION**

#### **COLLABORATIVE DU RACHIS SCOLIOTIQUE ..... 55**

4.1	Description globale du serveur développé.....	55
4.1.1	Détails d'implémentation et principaux composants .....	55
4.1.2	Diagramme de classe du serveur .....	57
4.2	Module S3 .....	58
4.2.1	Reconstruction 3D du rachis d'un patient spécifique .....	58
4.2.2	Modèle biomécanique du rachis .....	59
4.2.3	Initialisation du module S3.....	62
4.2.4	Positions optimales d'implants chirurgicaux .....	63
4.3	Module de manœuvres chirurgicales.....	64
4.4	Interface réseau du serveur.....	65

4.4.1	Processus de connexion et d'initialisation .....	65
4.4.2	Processus d'interaction pour l'instrumentation .....	66
4.4.3	Format et structure des messages .....	69
<b>CHAPITRE 5 : SERVEUR DE TÉLÉPRÉSENCE MULTI-USAGERS.....</b>		<b>71</b>
5.1	Description globale du serveur développé.....	71
5.1.1	Détails d'implémentation et principaux composants.....	71
5.1.2	Diagramme de classe du serveur.....	73
5.2	Gestion réseautique du serveur.....	73
5.2.1	Initiation de la communication .....	73
5.2.2	Processus de transmission des données de téléprésence.....	73
5.2.3	Format et structure des données échangées .....	74
<b>CHAPITRE 6 : CLIENT DE SIMULATION DE CHIRURGIE SCOLIOTIQUE</b>		
<b>EN RÉALITÉ VIRTUELLE .....</b>		<b>75</b>
6.1	Description globale du client développé .....	75
6.1.1	Détails d'implémentation et principaux composants.....	75
6.1.2	Diagramme de classe du client.....	78
6.1.3	Description générale de la salle de chirurgie virtuelle.....	79
6.2	Gestion de l'interaction usager .....	84
6.2.1	Interaction en mode immersif .....	84
6.2.2	Interaction en mode non-immersif .....	85
6.3	Interface avec les objets virtuels.....	87
6.3.1	Sélection d'un objet .....	87
6.3.2	Positionnement des projecteurs de la lampe chirurgicale.....	88
6.3.3	Choix d'un implant à l'aide d'une manette de sélection .....	88
6.3.4	Insertion d'un implant .....	89
6.3.5	Modification de la position d'insertion d'un implant.....	90
6.3.6	Suppression d'un implant .....	91

6.3.7	Modifier les paramètres des vues alternatives .....	91
6.3.8	Modifier la courbure d'une tige .....	92
6.3.9	Méthodes d'interaction proposées pour les manœuvres de S3... ..	92
6.3.10	Méthodes d'interaction et manœuvres implantées .....	97
6.3.11	Temps de réponse aux actions de l'utilisateur.....	101

## **CHAPITRE 7 : SIMULATION DE L'INSTRUMENTATION DU RACHIS**

<b>SCOLIOTIQUE .....</b>		<b>103</b>
7.1	Usage individuel du simulateur.....	103
7.2	Usage collaboratif du simulateur .....	104
7.2.1	Entraînement collaboratif.....	104
7.2.2	Télé-enseignement.....	105
7.3	Tests usagers .....	105
7.3.1	Description des participants .....	106
7.3.2	Mise en place et déroulement des tests .....	106
7.3.3	Résultats des tests .....	108

## **CHAPITRE 8 : DISCUSSION .....**

8.1	Limites et bénéfices.....	114
8.1.1	Serveur biomécanique d'instrumentation collaborative du rachis scoliotique .....	114
8.1.2	Serveur de téléprésence multi-usagers .....	115
8.1.3	Client de simulation de chirurgie scoliotique en réalité virtuelle. ....	116
8.1.4	L'ensemble du système.....	117
8.2	Avenues de recherches futures .....	118
8.2.1	Adaptation avancée du modèle biomécanique.....	119
8.2.2	Amélioration de l'interface non-immersive .....	119
8.2.3	Utilisation de gants de réalité virtuelle .....	120
8.2.4	Systèmes haptiques .....	120

8.2.5	Reconnaissance vocale.....	121
<b>CONCLUSION.....</b>		<b>122</b>
<b>RÉFÉRENCES .....</b>		<b>124</b>
<b>ANNEXES.....</b>		<b>130</b>

## Liste des annexes

<b>ANNEXE I : Diagramme de classe du serveur biomécanique et explications .....</b>	<b>130</b>
<b>ANNEXE II : Description des fonctions du module de manœuvres chirurgicales.....</b>	<b>133</b>
<b>ANNEXE III : Structures des messages réseaux envoyés ou reçus par le serveur biomécanique.....</b>	<b>137</b>
<b>ANNEXE IV : Exemples de messages réseaux envoyés ou reçus par le serveur biomécanique .....</b>	<b>143</b>
<b>ANNEXE V : Diagramme de classe du serveur de téléprésence et explications.....</b>	<b>147</b>
<b>ANNEXE VI : Structures des messages réseaux envoyés ou reçus par le serveur de téléprésence .....</b>	<b>149</b>
<b>ANNEXE VII : Diagramme de classe du client de simulation et explications.....</b>	<b>150</b>
<b>ANNEXE VIII : Formulaire d'évaluation .....</b>	<b>153</b>



## Liste des tableaux

Tableau 2.1 : Types de données transmises dans un EVC .....	30
Tableau 2.2 : Types de diffusion réseautique .....	31
Tableau 2.3 : Types de topologie réseautique couramment utilisés dans les EVC. ....	31
Tableau 4.1 : Positions des points d'insertion des implants .....	64
Tableau 4.2 : Requêtes d'instrumentation gérées par l'interface réseau du serveur biomécanique.....	66
Tableau 6.1 : Commandes de contrôle de la tête en mode non-immersif.....	86
Tableau 6.2 : Commandes de contrôle de la souris 3D en mode non-immersif .....	86
Tableau 6.3 : Méthodes d'attachement proposées pour CVrS3. ....	93
Tableau 6.4 : Temps de réponses des tâches d'instrumentation.....	101
Tableau II.1 : Description des fonctions de manœuvres chirurgicales de CAdamsManager de .....	133

## Liste des figures

Figure 1.1 : Comparaison entre une colonne vertébrale normale et scoliotique.	2
Figure 1.2 : Étapes de la chirurgie selon la technique Cotrel-Dubousset.....	4
Figure 1.3 : Insertion de vis pédiculaires.....	5
Figure 1.4 : Instrumentations proposées par six chirurgiens pour un même patient .....	6
Figure 1.5 : Modèles synthétiques .....	7
Figure 1.6 : Système de guidage StealthStation de Medtronic .....	9
Figure 1.7 : Captures d'écran du logiciel S3 .....	11
Figure 1.8 : Images de VRS3.....	12
Figure 2.1 : Dispositifs de réalité virtuelle .....	19
Figure 2.2 : Équipements externes de la CAVE.....	20
Figure 2.3 : Quelques domaines d'application de la RV .....	22
Figure 2.4 : Outils d'apprentissage et d'entraînement médicaux connus.....	25
Figure 2.5 : EVC à grande échelle connus .....	33
Figure 2.6 : Quelques EVC spécialisés connus .....	35
Figure 2.7 : Le projet TOUCH .....	38
Figure 2.8 : Architecture réseau du simulateur collaboratif de chirurgie de la cataracte .....	40
Figure 2.9 : Simulateur collaboratif de chirurgie de l'os temporal.....	41
Figure 3.1 : Architecture globale de CVrS3.....	44
Figure 3.2 : Console du serveur biomécanique .....	46
Figure 3.3 : Captures d'écran du client CVrS3 en mode non-immersif .....	48
Figure 3.4 : CVrS3 en mode immersif.....	49
Figure 3.5 : CVrS3 en mode non-immersif.....	50
Figure 3.6 : Modèles géométriques des implants de S3 .....	54
Figure 4.1 : Composantes du serveur biomécanique d'instrumentation collaborative du rachis scoliotique .....	56

Figure 4.2 : Capture d'écran du logiciel Spine 3D.....	59
Figure 4.3 : Modèle VRML de la colonne vertébrale d'un patient.....	63
Figure 5.1 : Composantes du serveur de téléprésence multi-usagers .....	72
Figure 6.1 : Couches logicielles du client de simulation de chirurgie scoliotique en réalité virtuelle.....	76
Figure 6.2 : Composantes du client de simulation de chirurgie scoliotique en réalité virtuelle .....	77
Figure 6.3 : Vues supplémentaires de la salle de chirurgie virtuelle .....	80
Figure 6.4 : Modèles géométriques de la salle de chirurgie virtuelle.....	83
Figure 6.5 : Avatars des participants .....	84
Figure 6.6 : Schéma de la souris 3D et de ses boutons.....	85
Figure 6.7 : Rotation de la souris 3D à appliquer pour déplacer un projecteur de la lampe .....	88
Figure 6.8 : Clic du bouton « précédent » de la manette de sélection de crochets.....	89
Figure 6.9 : Insertion d'un implant.....	90
Figure 6.10 : Modification de la courbure d'une tige .....	92
Figure 6.11 : Deux tentatives d'attachement de tige .....	98
Figure 6.12 : Déplacement de la souris 3D à appliquer pour une rotation de tige .....	99
Figure 6.13 : Application d'une manœuvre de rotation de tige.....	100
Figure 7.1 : Résultats du formulaire d'évaluation : réalisme géométrique.....	109
Figure 7.2 : Résultats du formulaire d'évaluation : interfaces .....	110
Figure 7.3 : Résultats du formulaire d'évaluation : immersion .....	111
Figure 7.4 : Résultats du formulaire d'évaluation : réalisme des tâches chirurgicales .....	112
Figure 7.5 : Résultats du formulaire d'évaluation : collaboration.....	112
Figure 7.6 : Résultats du formulaire d'évaluation : utilité.....	113
Figure I.1 : Diagramme de classe du serveur biomécanique .....	130

Figure V.1 : Composantes du serveur de téléprésence multi-usagers.....	147
Figure VI.1: Diagramme de classe du client de simulation de chirurgie scoliotique en réalité.....	150

## Liste des sigles et abréviations

2D	Deux dimensions, bidimensionnel.
3D	Trois dimensions, tridimensionnel.
API	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface, interface de programmation d'application.
CAO	<b>C</b> hirurgie <b>A</b> ssistée par <b>O</b> rdinateur.
CAVE	<b>C</b> ave <b>A</b> utomatic <b>V</b> irtual <b>E</b> nvironment, environnement immersif de réalité virtuelle s'apparentant à une cave.
CELTS	<b>C</b> omputer <b>E</b> nhanced <b>L</b> aparoscopic <b>T</b> raining <b>S</b> ystem, système d'entraînement à la chirurgie laparoscopique.
CHU	<b>C</b> entre <b>H</b> ospitalier <b>U</b> niversitaire.
CVrS3	<b>C</b> ollaborative <b>V</b> irtual reality <b>S</b> pine <b>S</b> urgery <b>S</b> imulator, le prototype de simulateur collaboratif de chirurgie scoliotique développé dans le cadre de cette étude.
DVD	<b>D</b> irect <b>V</b> ertebral <b>D</b> erotation, manœuvre de dérotation vertébrale directe.
EV	<b>E</b> nvironnement <b>V</b> irtuel
EVC	<b>E</b> nvironnement <b>V</b> irtuel de <b>C</b> ollaboration.
FIFO	<b>F</b> irst <b>I</b> n <b>F</b> irst <b>O</b> ut, principe informatique décrivant une file dans laquelle les premiers éléments arrivés sont les premiers traités.
HMD	<b>H</b> ead <b>M</b> ounted <b>D</b> isplay, visiocasque (casque de réalité virtuelle).
L1	Première vertèbre lombaire, située sous la dernière vertèbre thoracique (T12).
L5	Dernière vertèbre lombaire, située au-dessus de la première vertèbre sacrée.
LAT	Vue latérale.

LIS3D	Laboratoire Informatique de Scoliose <b>3D</b> .
LMBCAO	Laboratoire de <b>M</b> odélisation <b>B</b> iomécanique et de <b>C</b> hirurgie Assistée par <b>O</b> rdinateur.
Immersadesk	Environnement semi-immersif de réalité virtuelle commercialisé par la compagnie Fakespace.
IP	<b>I</b> nternet <b>P</b> rotocol.
m	Mètre, unité de distance.
mm	Millimètre, unité de distance.
Matlab	Un logiciel de calcul matriciel à syntaxe simple qui peut aussi être considéré comme un langage de programmation adapté pour les problèmes scientifiques.
MIST-VR	<b>M</b> inimally <b>I</b> nvasive <b>S</b> urgery <b>T</b> rainer – <b>V</b> irtual <b>R</b> eality, système d'entraînement et d'évaluation pour la chirurgie laparoscopique minimalement invasive.
N	Newton, unité de mesure de force.
Nmm	Unité de mesure de moment.
NCSA/AG	<b>N</b> ational <b>C</b> omputational <b>S</b> cience <b>A</b> lliance <b>A</b> ccess <b>G</b> rid.
PA	Vue antéropostérieure.
PC	<b>P</b> ersonal <b>C</b> omputer, ordinateur personnel.
RV	<b>R</b> éalité <b>V</b> irtuelle.
S3	<b>S</b> pine <b>S</b> urgery <b>S</b> imulator : simulateur de chirurgie scoliotique pour ordinateur conventionnel développé au LMBCAO.
SDK	<b>S</b> oftware <b>D</b> evelopment <b>K</b> it, kit de développement logiciel.

SGI	<b>Silicon Graphics International</b> , superordinateur surtout utilisé dans le domaine de la visualisation. Il s'agit également du nom de l'entreprise le fabriquant.
T1	Première vertèbre thoracique.
T10	Dixième vertèbre thoracique.
T11	Onzième vertèbre thoracique.
T12	Dernière vertèbre thoracique, située au-dessus de la première vertèbre lombaire (L1).
TCP	<b>Transmission Control Protocol</b> , protocole de contrôle de transmission.
UDP	<b>User Datagram Protocol</b> , protocole de datagramme utilisateur.
VRML	<b>Virtual Reality Modeling Language</b> , langage de description d'objets en réalité virtuelle.
VRS3	<b>Virtual Reality Spine Surgery Simulator</b> , simulateur de chirurgie scoliotique en réalité virtuelle. Il s'agit du prédécesseur de CVrS3 qui a été développé dans le cadre de la maîtrise d'Alexandre Plouznikoff.
Wand	Unité de contrôle tridimensionnelle (souris 3D) munie d'un presseur analogue (« joystick »), d'une gâchette et de deux boutons. Elle permet à l'utilisateur d'interagir à l'intérieur d'un environnement virtuel (tels que la CAVE et l'Immersadesk).

# CHAPITRE 1 : INTRODUCTION

## 1.1 Motivation de la recherche

Les patients atteints d'une scoliose idiopathique sévère doivent subir une intervention d'instrumentation chirurgicale hautement invasive et risquée. Pourtant, il n'existe aucun outil informatique d'entraînement collaboratif qui simule fidèlement l'expérience d'une telle chirurgie et qui permet une planification préopératoire pour un patient donné. L'objectif de cette recherche est d'évaluer la faisabilité du développement d'un tel simulateur à l'aide de dispositifs de réalité virtuelle (CAVE<sup>1</sup> et Immersadesk<sup>2</sup>), d'une modélisation biomécanique de la colonne vertébrale et de fonctionnalités collaboratives. Elle vise également à déterminer si un tel outil pourrait être utilisé efficacement pour l'entraînement des chirurgiens orthopédistes lors de sessions de télé-enseignement. Ces objectifs nécessitent la création d'un prototype (CVrS3<sup>3</sup>) simulant une salle de chirurgie virtuelle où plusieurs chirurgiens, géographiquement éloignés les uns des autres, pourront mutuellement se voir et collaborer lors de sessions d'entraînement à la chirurgie d'instrumentation du rachis scoliotique. Avant d'élaborer un peu plus quant aux éléments de problématique et aux objectifs de recherche d'une telle étude, une description succincte de la scoliose idiopathique, des traitements utilisés pour palier aux problèmes qu'elle peut engendrer et des outils d'entraînement et de planification chirurgicaux actuels est présentée.

---

<sup>1</sup> Cave Automatic Virtual Environment : un environnement immersif de réalité virtuelle.

<sup>2</sup> Environnement semi-immersif de réalité virtuelle.

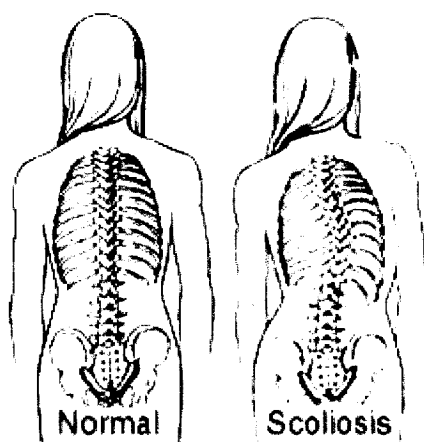
<sup>3</sup> Collaborative Virtual Reality Spine Surgery Simulator.



### 1.1.1 Définition de la scoliose idiopathique

La scoliose est définie comme une déformation tridimensionnelle partielle ou totale de la colonne vertébrale. Dubousset propose par ailleurs une définition formelle plus complète :

« Déformation rachidienne consistant en un déplacement relatif progressif d'un élément constitutif (vertèbre) par rapport à son adjacent, se produisant dans les trois plans de l'espace (frontal, sagittal et coronal) sans perte de la continuité ostéoligamentaire et se développant sur tout ou partie de la colonne vertébrale, essentiellement pendant la période de croissance ».



**Figure 1.1** : Comparaison entre une colonne vertébrale normale et scoliotique [SRS 2000].

Il est estimé que 10% des adolescents en sont atteints à un certain degré, mais que seulement 1% d'entre eux nécessitent une attention médicale [Scoliosis Association]. Quoiqu'elle atteigne aussi les hommes, elle touche plus fréquemment la population féminine qui est d'ailleurs plus fortement sujette à une progression nécessitant une intervention chirurgicale. Le terme « idiopathique » est utilisé lorsque la cause ne peut être déterminée, soit pour 80% à 90% des cas [NIAMS 2001]. Les 10% à 20% des cas restants sont attribuables à des anomalies congénitales, à des dérèglements neurologiques, à des conditions génétiques et à une multitude d'autres causes [SRS 2000].

### 1.1.2 Traitements de la scoliose

La sévérité de la scoliose et l'âge du patient constituent les principaux facteurs pris en compte lors de la préconisation d'un traitement. Tous les traitements utilisent une approche mécanique. Une scoliose légère peut généralement être corrigée à partir de mouvements de kinésithérapie. Lorsque cette méthode s'avère insuffisante, une solution un peu plus contraignante, quoique non invasive, est proposée : le port d'un corset. Ces deux méthodes ont pour objectif d'arrêter l'évolution de la déformation scoliotique. Lorsque qu'elles s'avèrent insuffisantes et que le caractère évolutif de la scoliose est important et mal contrôlé, une intervention chirurgicale est nécessaire. Plus précisément, cette approche invasive est préconisée si :

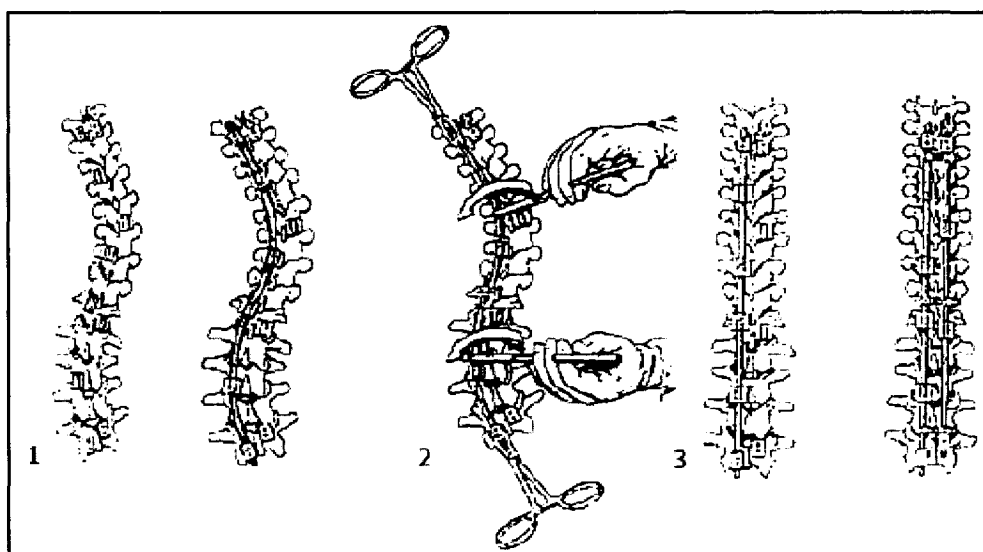
- La courbure progresse malgré l'utilisation d'un corset ;
- La courbure atteint 40 degrés en période de croissance ;
- La courbure dépasse 50 degrés et continue d'augmenter chez un adulte ;
- La douleur est difficilement tolérable.

La chirurgie consiste à instrumenter et redresser, via différentes manœuvres, la colonne vertébrale du patient afin de lui redonner une courbure approchant la normalité à l'aide de tiges métalliques fixées au rachis par des vis et des crochets. Une chirurgie scoliotique typique (figure 1.2) est généralement constituée des étapes suivantes [Lonstein 1995] :

#### 1. Instrumentation de la colonne vertébrale :

- Positionnement et insertion d'implants vertébraux (vis, crochets) aux endroits jugés appropriés ;
- Fixation d'une première tige (préalablement déformée selon la courbure désirée) aux implants ;

2. Application de manœuvres correctrices (rotation de tige, translation, compression, distraction et/ou dérotation directe de vertèbres, ...) à l'aide d'outils spécialisés ;
3. Fixation d'une deuxième tige afin de solidifier et conserver les transformations préalablement appliquées ;
4. Fusion des vertèbres au moyen d'une greffe osseuse afin de solidifier l'instrumentation.



**Figure 1.2 :** Étapes de la chirurgie selon la technique Cotrel-Dubousset [Lonstein 1995].

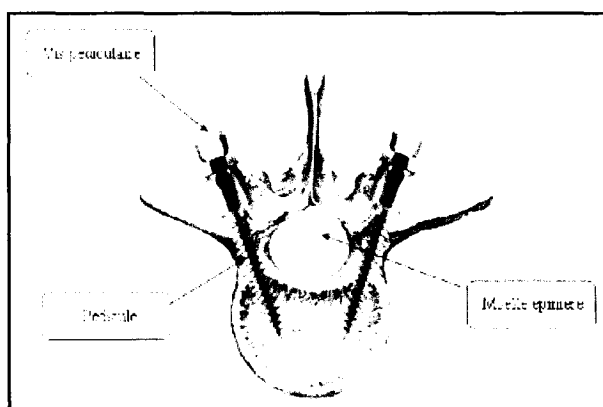
Bien que le simulateur développé dans le cadre de cette étude soit basé sur cette technique, il est important de remarquer qu'il existe d'autres approches d'intervention.

La planification préopératoire, pour une chirurgie scoliotique, repose principalement sur l'analyse des projections radiographiques postéro-antérieures et sur des tests d'inflexion latérale qui permettent de caractériser les courbures rachidiennes, d'évaluer leur flexibilité et leur potentiel de correction. Certaines informations cliniques, telles que l'âge et le sexe du patient, sont également considérées. La plupart des chirurgiens utilisent encore aujourd'hui cette méthode conventionnelle.

### 1.1.3 Outils chirurgicaux

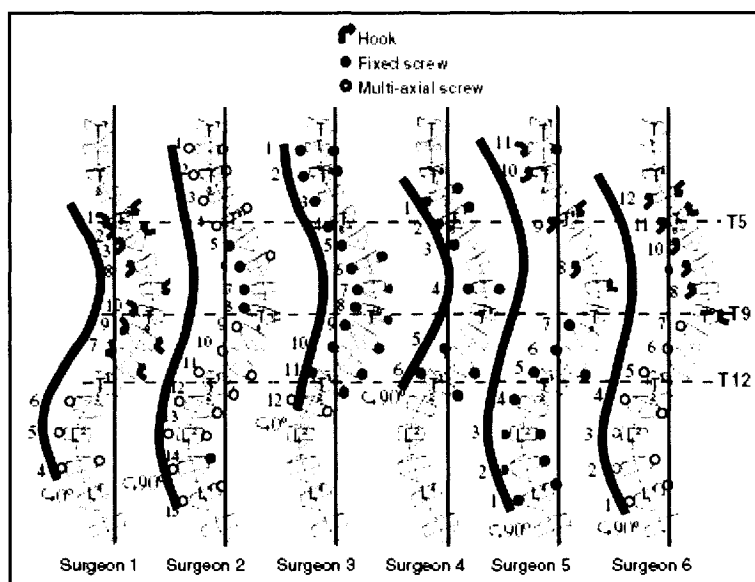
Les risques inhérents à la chirurgie scoliotique sont nombreux. Les plus communs sont :

- La paralysie des membres inférieurs causée par une lésion de la moelle épinière lors de l'insertion de vis pédiculaires (figure 1.3) ;
- L'arrachage des implants et la fracture des vertèbres lors de l'application d'une force de correction excessive.



**Figure 1.3 :** Insertion de vis pédiculaires (Spine-Health).

En plus des risques impliqués, il faut préciser qu'il s'agit d'une procédure complexe impliquant plusieurs décisions difficiles (le choix du segment spinal à instrumenter, le type, la position et le nombre de vis ou de crochets à insérer, le diamètre, la longueur et la forme de la tige, l'ordre d'attachement des implants, la quantité de rotation à appliquer sur une tige, ...). Les récentes innovations en matière de technologie d'instrumentation ont amené une forte augmentation des options chirurgicales. Malgré ces innovations et le fait qu'il existe plusieurs publications cliniques pour guider l'intervention, il n'existe aucun consensus parmi les chirurgiens quant au plan optimal à utiliser pour chaque type de courbure scoliotique [Aubin et al. 2006, 1]. Il existe d'ailleurs une grande variabilité entre les solutions adoptées, par différents chirurgiens, à un même problème orthopédique (figure 1.4).



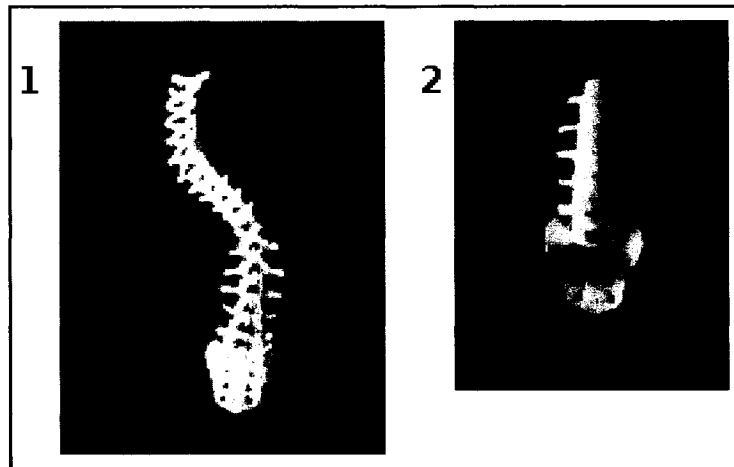
**Figure 1.4 :** Instrumentations proposées par six chirurgiens pour un même patient [Aubin et al. 2006, 1].

C'est afin de minimiser ces risques, d'améliorer le pronostic et de maximiser l'impact thérapeutique de l'intervention que plusieurs outils d'entraînement et de planification chirurgicaux ont été développés.

### Outils d'entraînement chirurgical

Outre la pratique sur patients où un médecin résident peut, par exemple, assister un chirurgien expérimenté, il existe deux alternatives pour l'entraînement : les modèles synthétiques et la pratique sur cadavres.

Les modèles synthétiques (figure 1.5) sont généralement constitués de bois, de plastique ou de polystyrène. Des reproductions complètes ou partielles de la colonne sont disponibles. Il est possible de choisir le matériel osseux, d'obtenir une reproduction avec ou sans pathologie, avec ou sans réponse mécanique. Le choix du matériel osseux a un impact direct sur le degré de réalisme des forces ressenties lors de l'insertion d'une vis dans un pédicule.



**Figure 1.5 :** Modèles synthétiques : 1- Colonne vertébrale complète avec courbe scoliotique gauche de 55 degrés ; 2- Colonne vertébrale lombaire (L1 à L5 avec sacrum) simulant les ligaments longitudinaux antérieurs et postérieurs (Sawbones).

La pratique sur cadavres constitue une option qui offre plus de réalisme et qui procure une meilleure sensation d'immersion dans la simulation. Elle possède néanmoins plusieurs inconvénients :

- Son coût est de plus en plus élevé. L'accès aux cadavres est de plus en plus encadré et les lois les régulant sont de plus en plus strictes ;
- Parmi le peu de cadavres disponibles, il est très rare de retrouver des cas scoliotiques. La simulation est souvent contrainte à une colonne vertébrale saine dont l'anatomie et la biomécanique ne peuvent être choisies. Les cadavres avec scoliose sont la plupart du temps ceux d'adultes ou de personnes âgées ;
- Les propriétés physiques et biomécaniques de tissus morts ne sont plus fidèles à celles qu'on obtient avec un spécimen vivant.

Ces outils d'entraînement sont aujourd'hui principalement utilisés pour le parachèvement de la formation pratique des médecins résidents en orthopédie, mais une nouvelle préoccupation s'impose de plus en plus : l'importance de l'apprentissage à vie. C'est afin d'assurer un standard de qualité d'excellence que R. Aggarwal et ses collaborateurs proposent un programme d'entraînement chirurgical progressif faisant

partie d'un projet national [Aggarwal et al. 2004]. Un tel programme devrait être conçu de façon à s'adapter au niveau de compétence du chirurgien : les médecins résidents l'utiliseraient comme outil d'initiation tandis que les vétérans l'utiliseraient afin de spécialiser et perfectionner leurs techniques.

### **Outils de planification et d'assistance chirurgicales**

Plusieurs techniques d'imagerie 3D ont été développées afin de mieux comprendre la nature tridimensionnelle de la scoliose idiopathique et d'améliorer le processus de planification chirurgicale.

La stéréophotogrammétrie ou la reconstruction 3D radiographique est à la base de plusieurs travaux de recherche. Ce procédé d'analyse d'image permet d'obtenir les coordonnées 3D associées aux coordonnées 2D des radiographies antéropostérieures et latérales du rachis. Il offre l'avantage d'être facilement intégrable en salle clinique et de fournir une représentation géométrique détaillée des structures osseuses internes qui est adéquate pour l'évaluation clinique des difformités scoliotiques 3D régionales et globales [Petit et al. 2002]. La stéréophotogrammétrie est malgré tout très peu utilisée en contexte clinique.

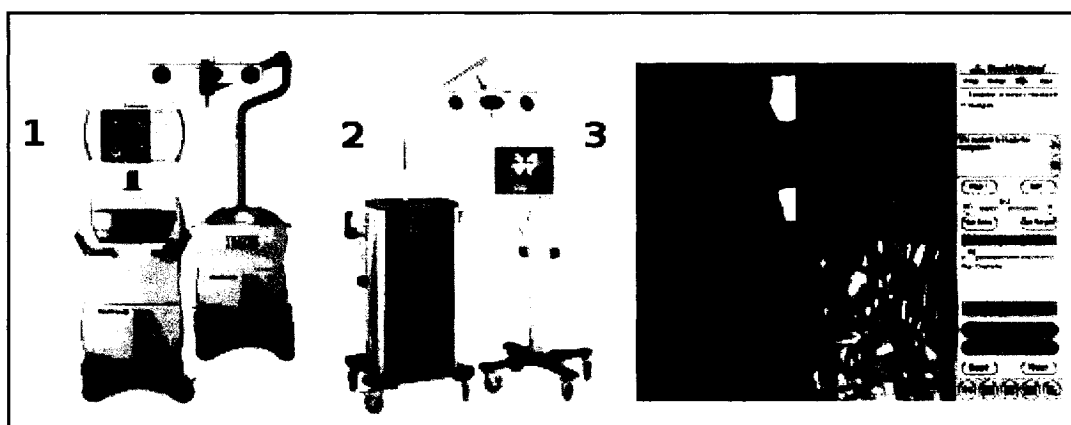
La topographie de surface permet la quantification des difformités 3D de surface du tronc scoliotique. Il est possible, à l'aide de dispositifs d'imagerie, d'obtenir une reconstruction 3D des formes de surface. Cette représentation permet l'étude 3D des difformités externes du tronc scoliotique et est complémentaire aux techniques radiographiques puisqu'elle s'associe plus facilement aux difformités telles que perçues par le patient.

La tomographie assistée par ordinateur et l'imagerie par résonance magnétique sont un peu plus utilisées en contexte clinique. La tomographie est utilisée pour générer un ensemble d'images détaillées de l'anatomie osseuse. L'imagerie par résonance magnétique génère, quant à elle, des images détaillées des tissus mous. Étant donné leurs

grandes précisions, ces techniques sont idéales pour la quantification des difformités intrinsèques. Elles demeurent malgré tout non idéales pour visualiser les changements globaux de l'ensemble de la colonne. L'imagerie par résonance magnétique est un long processus et produit, une fois la chirurgie terminée, des artéfacts dus aux implants métalliques. La tomographie produit des images adéquates en pré et post opération, mais le haut niveau de radiation auquel on expose le patient oblige une utilisation régionale.

Bien que peu utilisés par les chirurgiens, des outils d'assistance intraopératoire sont également disponibles. Le système Navitrack Spine de la compagnie ORTHOsoft est l'un d'entre eux. Il s'agit d'un système de guidage pour la pose de vis pédiculaires qui repère les implants, les instruments et les vertèbres afin de générer en temps réel une ou plusieurs vues tridimensionnelles sur la zone d'intérêt.

Le système StealthStation (figure 1.6) de Medtronic poursuit un objectif semblable. Le modèle Treon combine des images préopératoires, généralement générées par imagerie par résonance magnétique et tomographie, avec les images intraopératoires. Le chirurgien peut contrôler la position des caméras afin d'obtenir la vue la plus précise possible de la zone concernée. Le modèle Ion utilise, quant à lui, un système de fluoroscopie virtuelle afin d'obtenir une vue intraopératoire détaillée de l'anatomie du rachis.



**Figure 1.6:** Système de guidage StealthStation de Medtronic :  
1- Le modèle Treon ; 2- Le modèle Ion ; 3-Écran de navigation.



## 1.2 Contexte de la recherche

Le présent travail est un projet de recherche du Laboratoire de Modélisation Biomédicale et de Chirurgie Assistée par Ordinateur (LMBCAO). Ce laboratoire, partagé par l'École Polytechnique de Montréal et le Centre de recherche du CHU<sup>4</sup> Sainte-Justine, a pour mission de mieux comprendre les mécanismes pathologiques et la biomécanique de la colonne vertébrale afin de trouver des traitements pour la scoliose d'une efficacité optimale. Ce projet est supporté par la Chaire de recherche du Canada Innovations CAO en génie orthopédique qui vise à améliorer le traitement des déformations musculosquelettiques de la colonne vertébrale en concevant des outils informatiques, des modèles de simulations et des techniques chirurgicales minimalement invasives de correction assistée par ordinateur.

Le prototype développé dans le cadre de cette étude, CVrS3 (Collaborative Virtual Reality Spine Surgery Simulator), constitue une prolongation du logiciel S3 (Spine Surgery Simulator) : un simulateur de chirurgie scoliotique pour ordinateurs avec interface conventionnelle (figure 1.7). Ce puissant logiciel 3D, qui est le fruit de nombreuses années de travail de chercheurs du LMBCAO, a comme objectif de procurer aux chirurgiens orthopédistes un outil avec lequel il est possible de tester rapidement, avant la réelle chirurgie, différentes stratégies d'intervention pour un patient donné. La colonne vertébrale d'un patient donné est modélisée, géométriquement, à partir de deux radiographies calibrées et, biomécaniquement, à partir d'informations concernant sa flexibilité. Il est donc possible, à l'aide de menus, d'insérer virtuellement des implants sur la reproduction 3D de la colonne du patient et d'effectuer différentes manœuvres correctrices pour finalement observer l'impact de l'intervention sur le rachis. Les arguments justifiant la création d'une version en réalité virtuelle seront présentés dans la prochaine section.

---

<sup>4</sup> Centre Hospitalier Universitaire

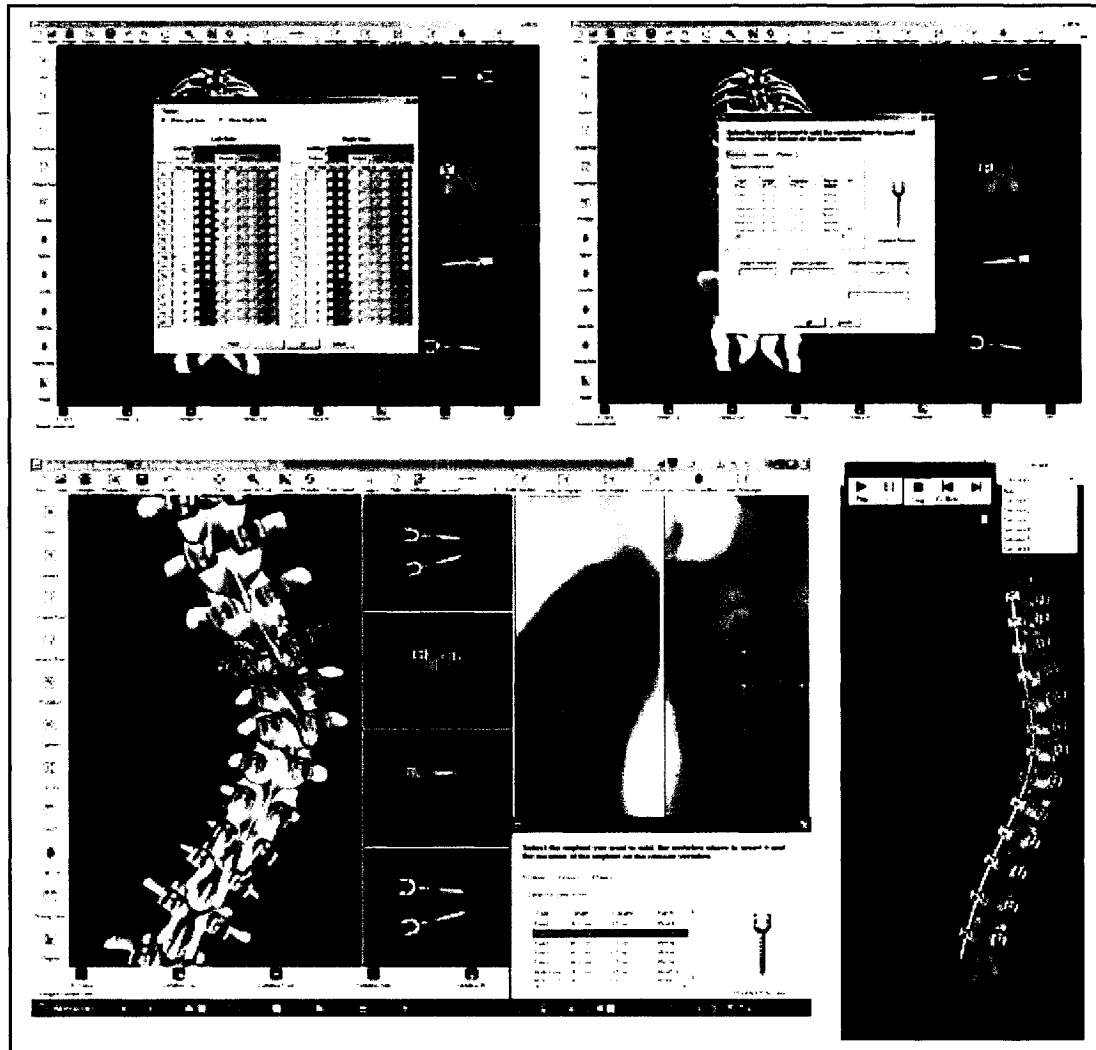
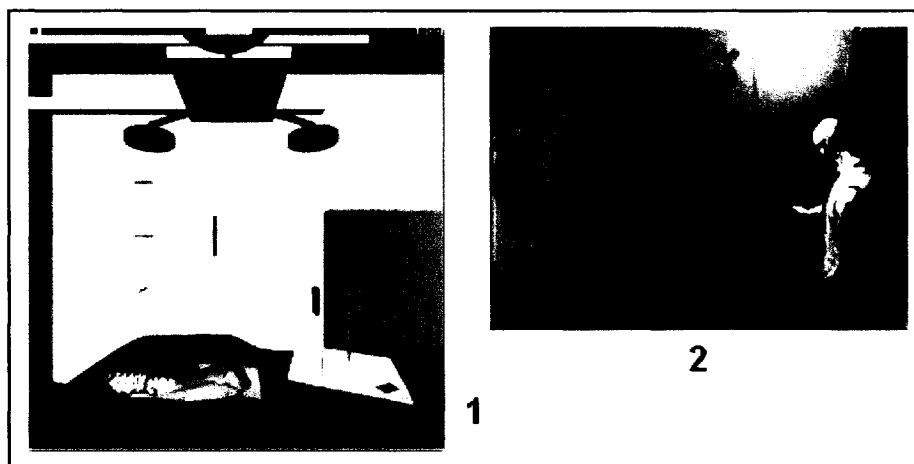


Figure 1.7 : Captures d'écran du logiciel S3.

Bien qu'il s'agisse bel et bien d'un prototype distinct, il est important de mentionner que CVrS3 est le successeur de VRS3 (Virtual Reality Spine Surgery Simulator) (figure 1.8). Ce dernier, développé par Alexandre Plouznikoff dans le cadre de sa maîtrise, constitue le premier prototype fonctionnel d'un simulateur de chirurgie de troisième génération permettant de simuler l'instrumentation du rachis scoliotique, tant d'un point de vue graphique que biomécanique [Plouznikoff 2004]. VRS3 est conçu pour un système de réalité virtuelle de type CAVE (Cave Automatic Virtual Environment). Cet environnement permet, à l'aide de projecteurs et d'écrans, d'un système de repérage, de

lunettes stéréoscopiques et d'une souris 3D (« wand »), d'immerger l'utilisateur dans une salle de chirurgie virtuelle interactive. Plus d'informations à propos de la CAVE seront présentées au début du deuxième chapitre.



**Figure 1.8** : Images de VRS3 : 1- Capture d'écran du simulateur en mode simulation sur un ordinateur avec écran conventionnel ; 2- Photo d'un utilisateur s'entraînant avec VRS3 dans la CAVE (vue monoscopique).

CVrS3 s'inspire de VRS3, mais a dû être réimplanté à partir de zéro puisqu'il utilisait une vieille version, peu réaliste du moteur biomécanique de S3, qui n'offrait pas l'éventail des manœuvres chirurgicales calquées sur le réel qu'offre aujourd'hui la plus récente version de S3. Le fait qu'il était conçu avec la librairie CAVELib pour fonctionner avec un superordinateur SGI (Silicon Graphics International) très coûteux, peu récent et dépassé en terme de performances<sup>5</sup>, a également motivé ce choix. D'un point de vue général, CVrS3 se distingue de son prédécesseur par l'idée directrice suivante : la volonté de créer une version **collaborative** du simulateur en réalité virtuelle. Les arguments justifiant la création d'une version collaborative seront présentés dans la prochaine section.

---

<sup>5</sup> En comparaison avec la grappe de PC, équipés de cartes graphiques spécialisées, utilisées actuellement pour l'environnement de la CAVE de l'École Polytechnique.

### 1.3 Éléments de problématique

Voici les problèmes soulignés par Alexandre Plouznikoff quant aux méthodes d'entraînement traditionnel [Plouznikoff 2004] :

« Les outils d'entraînement actuels, bien que très utiles pour compléter la formation des médecins résidents, ne répondent que partiellement aux attentes de la communauté universitaire car :

- Anatomiquement, les outils d'entraînement sont de loin trop généraux. En effet, il est difficile d'obtenir des os synthétiques simulant correctement un type de scoliose donné car ceux-ci sont fabriqués de manière non spécifique.
- Biomécaniquement, les outils d'entraînement présentent des réponses peu fidèles (dans le cas des os synthétiques) ou altérées (dans les rares cas de cadavres présentant une scoliose). Que ce soit pour simuler l'insertion d'une vis pédiculaire ou simuler les forces correctrices à exercer pour corriger une déformation scoliotique, ces outils ne simulent pas fidèlement les forces en jeu lors d'une réelle chirurgie.
- Économiquement, la pratique sur cadavre est très onéreuse vu la rareté des spécimens, la lourdeur des nouvelles régulations et les coûts en personnel et en matériel requis. Bien que moins coûteuse, la pratique sur os synthétiques est elle aussi relativement onéreuse, si bien qu'il est rare que chaque étudiant puisse pratiquer plusieurs fois sur de tels spécimens. Mentionnons aussi que chaque séance d'entraînement détruit les spécimens utilisés, augmentant de surcroît les coûts d'une telle formation ». Il est donc impossible d'effectuer plusieurs interventions sur un même spécimen afin de trouver la meilleure stratégie.
- « Quant aux outils de planification, bien qu'ils permettent actuellement d'améliorer de manière anatomique le pronostic des instrumentations du rachis scoliotique, ils ne répondent pas entièrement aux attentes de la communauté médicale car, biomécaniquement, aucun outil ne permet de planifier ou de simuler une pathologie spécifique à un patient afin de tester plusieurs instrumentations et de discerner celle aboutissant à la meilleure correction ».

VRS3 offre une solution à ces problèmes, mais ne procure pas une solution optimale aux deuxième et quatrième points puisque la version de S3 qui y est intégrée implémente une modélisation biomécanique de la colonne vertébrale (conçue avec l'outil Matlab)

simplifiée et peu réaliste. La plus récente version de S3 conçue à l'aide du logiciel et du SDK<sup>6</sup> Adams de MSC Software, qui permettent de modéliser et de simuler des systèmes mécaniques complexes, peut maintenant prétendre apporter une solution plus réaliste. L'utilisation d'un ordinateur avec écran conventionnel et d'une souris, afin d'interagir avec des menus tout au long du processus de simulation, fait de S3 un excellent outil préopératoire pour tester rapidement plusieurs scénarios d'opération. Cet avantage a cependant le désavantage suivant : l'expérience sensorielle procurée est très loin de ressembler à une vraie chirurgie. Les remarques suivantes précisent ce désavantage :

- Le sentiment d'immersion dans la simulation est très faible puisque la vue obtenue ne montre qu'une projection 2D de la colonne vertébrale du patient. Le corps du patient, la table et la salle d'opération ne sont pas affichés. L'utilisateur ne peut donc pas se déplacer à l'intérieur de la salle et observer autour de lui. L'absence d'une vue stéréoscopique contribue également au faible sentiment d'immersion ;
- Le fait de ne pas pouvoir utiliser ses deux mains, pour manipuler les outils et les implants chirurgicaux, pour ressentir la force appliquée sur la colonne vertébrale durant les manœuvres, diminue grandement le réalisme de la simulation.

VRS3 constitue une première approche de solution à ces problèmes, mais il possède néanmoins les limites suivantes :

- La variété des implants pouvant être utilisés est très limitée. Les chirurgiens utilisent normalement trois catégories d'implants (crochets, vis pédiculaires statiques et multiaxiales) à l'intérieur desquels plusieurs configurations sont offertes pour permettre l'instrumentation optimale du rachis d'un patient spécifique ;

---

<sup>6</sup> Software Development Kit : kit de développement logiciel.

- La variété des manœuvres chirurgicales implantées est limitée ;
- Il ne peut être utilisé que par une seule personne à la fois. Le fait que la chirurgie scoliotique nécessite, généralement, le travail coopératif de deux chirurgiens rend cet aspect encore plus problématique ;
- Il est uniquement conçu pour fonctionner avec un environnement de réalité virtuelle tel que la CAVE qui est très rare et coûteux ;
- Lors d'un contexte d'apprentissage, tous les participants doivent se retrouver au même endroit physique et ils sont contraints (sauf l'instructeur) à jouer le rôle d'observateurs.

## 1.4 Objectifs de recherche

L'objectif de cette étude est de créer un prototype (CVrS3) qui permet la collaboration à distance simultanée de plusieurs utilisateurs, pouvant mutuellement se voir et communiquer à l'aide d'un affichage commun, d'avatars et de transmission de voix, lors d'une session d'entraînement. Cet objectif implique qu'il faudra :

- Créer une salle de chirurgie virtuelle composée d'une lampe chirurgicale, d'étagères, d'un négatoscope (pour les radiographies), d'une table d'opération, d'un drap chirurgical, d'un plateau avec implants et outils, du corps d'un patient et d'une colonne vertébrale où il est possible de se déplacer et d'interagir avec les objets ;
- S'assurer que le simulateur ne fonctionne pas seulement dans un environnement de réalité virtuelle immersif (CAVE) ou semi-immersif (Immersadesk), mais aussi avec un ordinateur avec écran conventionnel, clavier et souris ;
- Intégrer le plus récent moteur biomécanique de la colonne vertébrale de S3 afin de pouvoir instrumenter la colonne d'un patient spécifique et effectuer des

manœuvres correctrices tout en observant en temps réel l'impact de celles-ci sur sa courbure ;

- Établir une architecture réseau client-serveur entre plusieurs instances du simulateur (les clients) et le moteur biomécanique (le serveur) qui conserve l'état et permet aux utilisateurs d'effectuer des requêtes sur une seule version, commune, de la colonne vertébrale d'un patient spécifique ;
- Établir une architecture client-serveur entre plusieurs instances du simulateur (les clients) et un serveur de télé-présence multi-usagers qui transmet les moindres actions de chaque usager (déplacement, sélection et déplacement d'objet, ...) à tous les autres ;
- Utiliser ou adapter un outil permettant une communication vocale entre tous les usagers ;
- Évaluer à l'aide de tests d'utilisabilité si l'ajout de fonctionnalités collaboratives au simulateur peut ou pourra être utile pour l'entraînement des chirurgiens orthopédistes.

## **1.5 Contributions et originalité des travaux**

Le American College Of Surgeons [Dawson 2002] classifie les simulateurs médicaux en trois générations. La première génération représente les simulateurs médicaux purement géométriques qui permettent d'observer et de se déplacer à proximité d'un organe ou d'une partie d'un corps humain virtuel. Ce type de simulateur peut être utilisé pour compléter la formation théorique des étudiants en médecine. La deuxième génération permet d'interagir de façon prédéterminée avec des organes et des tissus spécifiques du corps humain. Ceux-ci sont normalement utilisés pour l'entraînement des médecins résidents et leurs permettent de répéter des manœuvres spécifiques à plusieurs reprises.

La contribution majeure de ce projet est de créer le premier prototype d'un simulateur multi-usagers de troisième génération qui permet une collaboration à distance lors de la simulation graphique et biomécanique de la chirurgie scoliotique. Il n'existe d'ailleurs aucun autre simulateur en réalité virtuelle, qu'il soit conçu pour une utilisation individuelle ou collaborative, qui offre une modélisation biomécanique du rachis scoliotique aussi complexe et réaliste.

## **1.6 Organisation du mémoire**

Ce mémoire est constitué de huit chapitres. Ce premier chapitre introduisait la motivation, le contexte, les éléments de problématique et les objectifs de cette étude. La scoliose idiopathique, ses traitements et les principaux outils chirurgicaux y étaient décrits. Le deuxième chapitre est une revue de la littérature et des travaux antérieurs. Elle propose une description du domaine de la réalité virtuelle et des environnements virtuels de collaboration. Elle présente également des outils et simulateurs médicaux, appartenant à ces disciplines, qui méritent d'être mentionnés et/ou décrits quant à leurs similarités avec le prototype développé pour cette étude. Le troisième chapitre présente l'architecture globale du simulateur et il introduit ses principales entités : le serveur biomécanique d'instrumentation collaborative du rachis scoliotique, le serveur de téléprésence multi-usagers et le client de simulation de chirurgie scoliotique en réalité virtuelle. Les outils et données initiales utilisés pour l'élaboration du simulateur sont également présentés. Les chapitres quatre, cinq et six ont pour objectifs de décrire plus en détails les trois entités mentionnées en présentant les concepts inhérents à leur conception, leurs structures logicielles et certains détails d'implémentation. Le chapitre sept présente les scénarios d'utilisation individuelle et collaborative du simulateur. Il décrit aussi les aspects organisationnels et les résultats de l'étude d'utilisabilité effectuée pour le prototype développé. Finalement, le dernier chapitre présente une discussion dont l'objectif est de cerner les limites et les bénéfices du simulateur, et d'ouvrir la voie à des avenues de recherche futures.



## CHAPITRE 2 : REVUE DE LA LITTÉRATURE

Ce chapitre propose une revue de la littérature et des travaux antérieurs liés aux domaines de la réalité virtuelle (RV) et des environnements virtuels de collaboration (EVC). Certains des concepts incontournables de ces disciplines sont abordés et quelques-uns des outils et simulateurs médicaux en réalité virtuelle, des EVC à grande échelle et des EVC spécialisés pour le contexte éducatif et/ou médical sont présentés.

### 2.1 Outils et simulateurs médicaux en réalité virtuelle

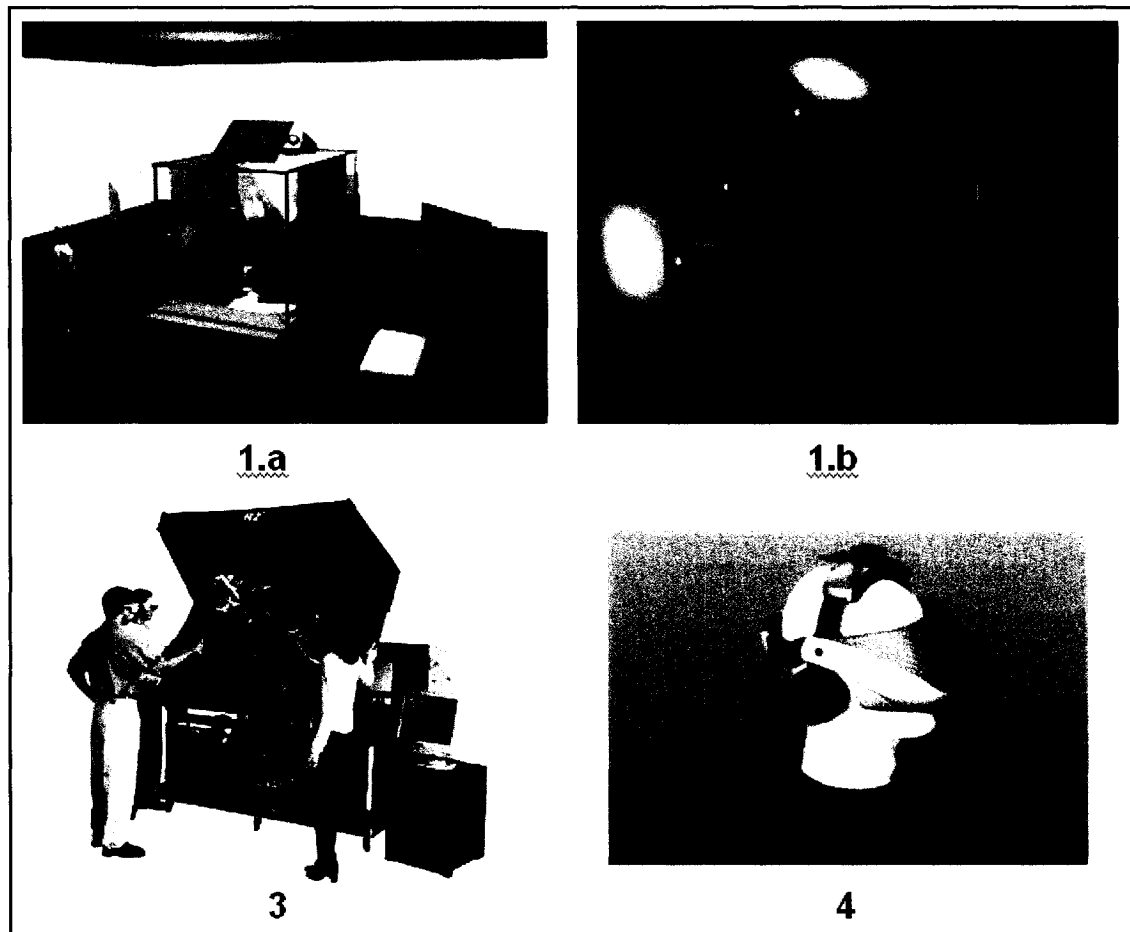
Avant de présenter certains des outils et simulateurs médicaux en réalité virtuelle connus, il est nécessaire de décrire succinctement le domaine de la réalité virtuelle et les dispositifs communément utilisés. Les domaines d'application de la réalité virtuelle et des exemples de logiciels divers seront également présentés.

#### 2.1.1 La réalité virtuelle : définition et caractéristiques

La réalité virtuelle est une simulation informatique d'environnements qui utilisent des images 3D et des équipements externes comme des gants de RV et des casques pour permettre à l'utilisateur d'interagir avec la simulation. Les utilisateurs se déplacent au travers des environnements de réalité virtuelle comme s'ils se déplaçaient dans le vrai monde, marchent à l'intérieur de structures et interagissent avec des objets de cet environnement en temps réel. La CAVE, l'Immersadesk et les visiocasques (« Head Mounted Display », HMD) sont des exemples concrets de dispositifs de réalité virtuelle (figure 2.1). Bien que la plupart des environnements de RV actuels se contentent de générer des images 3D et du son, il existe aussi des dispositifs spécialisés, dont l'efficacité dépend du contexte d'utilisation, pour simuler le toucher<sup>7</sup>, l'odorat et le goût.

---

<sup>7</sup> Les dispositifs simulant le toucher, la saisie d'objet ou le retour d'effort sont appelés systèmes haptiques.



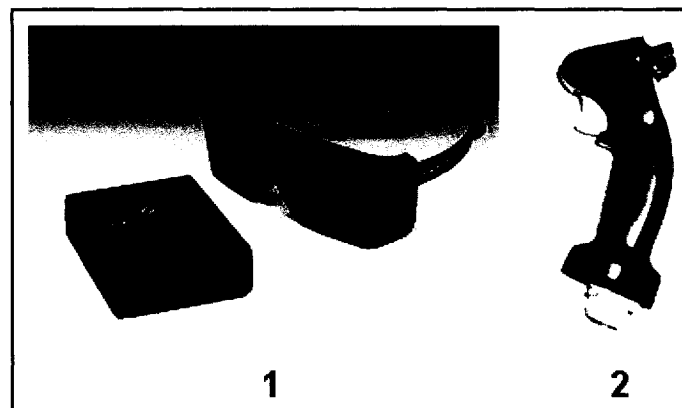
**Figure 2.1** : Dispositifs de réalité virtuelle : 1- a et b montrent la CAVE (Fakespace) ;  
2- L'Immersadesk (Fakespace) ; 3- Un visiocasque (5DT HMD 800 Series).

### **La CAVE et l'Immersadesk**

Puisque CVrS3 est conçu pour fonctionner avec la CAVE de l'École Polytechnique et l'Immersadesk du LMBCAO, une description de ces systèmes s'impose.

La CAVE (Cave Automatic Virtual Environment), appelée aussi « voûte d'immersion », est un environnement immersif de réalité virtuelle se présentant sous la forme de trois parois et d'un plancher disposés dans un cube de trois mètres de côté. Cet environnement a été présenté pour la première fois lors de la conférence SIGGRAPH'92, mais est dorénavant distribué par la compagnie Fakespace. Les trois parois et le plancher constituent quatre écrans sur lesquels il est possible de projeter, à l'aide de projecteurs

positionnés à l'arrière des parois et au dessus du plancher, des images de synthèse 3D calculées en temps réel. Les images stéréoscopiques projetées sur les écrans sont générées pour chaque œil de l'utilisateur et sont affichées en alternance afin que les lunettes stéréoscopiques (figure 2.2) puissent se synchroniser et obstruer la vision de l'œil gauche lorsque l'image pour l'œil droit est affichée et vice-versa. Un système de repérage, retournant la position de l'utilisateur et son orientation dans l'espace en temps réel, est connecté aux lunettes afin d'afficher des images de la scène générée par rapport à la perspective spécifique de l'usager à un moment donné. Il est donc possible de voir des objets en trois dimensions d'une manière extrêmement réaliste et de se déplacer autour de ceux-ci afin de les examiner sous n'importe quel angle. Une souris 3D (« wand ») (figure 2.2) ou des gants, également connectés à un système de repérage, sont normalement utilisés afin d'interagir avec la simulation. Le « wand » est une unité de contrôle munie d'un « joystick » (presseur analogue), d'une gâchette et de deux boutons permettant à l'utilisateur de se déplacer dans l'environnement virtuel et d'y faire des actions.



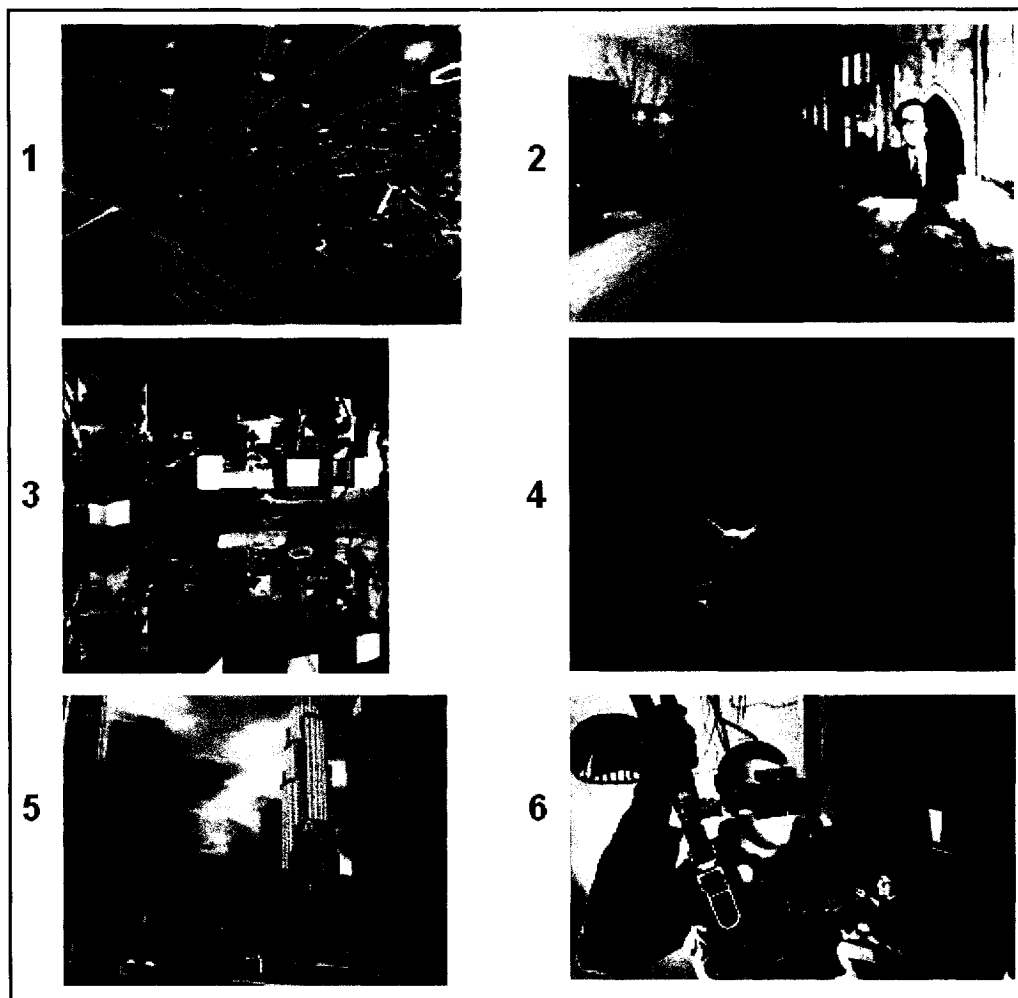
**Figure 2.2 :** Équipements externes de la CAVE :  
1- Lunettes stéréoscopiques (StereoGraphics CrystalEyes) ; 2- Wand (Fakespace).

L'Immersadesk est un environnement semi-immersif semblable à la CAVE qui ne comporte qu'un seul écran (d'environ 1m20 par 1m50). Il s'agit également d'un système distribué par la compagnie Fakespace. Bien qu'il soit possible de modifier l'angle de l'écran par rapport au sol pour donner une impression de profondeur, il faut tout de même remarquer que le type d'immersion procuré n'est pas aussi convaincant qu'avec

une CAVE. Il possède par contre l'avantage de pouvoir se replier complètement dans un caisson de façon à être facilement transportable et logeable.

### **2.1.2 Domaines d'application de la réalité virtuelle**

La réalité virtuelle suscite un fort intérêt chez les chercheurs et le grand public depuis plusieurs années. Les applications en réalité virtuelle possèdent généralement une très grande complexité. Une très forte puissance de calcul est donc nécessaire. La réalité virtuelle est considérée comme une technologie offrant d'innombrables possibilités, mais ce n'est que depuis peu de temps, grâce au moindre coût et à la performance des ordinateurs, des cartes graphiques et des équipements externes actuels, qu'il est enfin possible de créer des applications concrètement utilisables. La réalité virtuelle est aujourd'hui utilisée en recherche ou comme outil commercial dans plusieurs domaines (figure 2.3) tels que la visualisation scientifique, l'ingénierie, la vidéoconférence, les arts, le divertissement, l'architecture, le design, la médecine et le militaire [Ozell 2003].



**Figure 2.3 :** Quelques domaines d'application de la RV : 1- Visualisation scientifique ; 2- Vidéoconférence ; 3- Arts (Las Meninas) ; 4- Divertissement (Quake II) ; 5- Architecture et design (Downtown Los Angeles) ; 6- Entraînement militaire (US Navy parachute trainer).

### 2.1.3 Les outils et simulateurs médicaux en réalité virtuelle

La prise de conscience du milieu médical face aux avantages que peut apporter la réalité virtuelle pour l'amélioration de la qualité des soins prodigués aux patients est relativement récente [Zajtchuk et Satava 1997]. Son utilisation en médecine peut être séparée en cinq grandes catégories [Plouznikoff 2004] :

- |  |   |
|--|---|
| 1. Fusion de données intra-opératoires ;   | 2. Planification et simulation de désastres ; |
| 3. Prototypage virtuel ;                   | 4. Réadaptation et psychothérapie ;           |
| 5. Éducation et entraînement des médecins. |   |

La première catégorie fait référence aux systèmes de réalité augmentée, utilisés en salle opératoire, pour guider le chirurgien lors de l'intervention grâce à la superposition des modèles anatomiques 3D sur le corps du patient. Des recherches sont d'ailleurs en cours quant à l'utilisation de cette technologie pour l'instrumentation du rachis scoliotique. Peuchot et ses collaborateurs s'y intéressent notamment dans le but d'améliorer le guidage des vis pédiculaires [Peuchot et al. 1995] [Tanguy et Peuchot 2002].

La deuxième catégorie est plus générale et représente les systèmes de simulation d'événements critiques et de gestion des comportements à adopter en cas de désastres naturels, de crises, ...

La réalité virtuelle est de plus en plus utilisée pour le design et le prototypage de dispositifs mécaniques, d'automobiles, d'immeubles, ... Le domaine médical profite également de la minimisation des coûts qu'apporte le design virtuel préalable à la fabrication matérielle de prototypes ou du produit final. La RV est utilisée pour le prototypage de nouveaux implants, pour tester virtuellement des nouveaux concepts de salles d'opération et d'équipements.

Le quatrième point constitue un domaine de recherche qui jouit d'une popularité grandissante. La réalité virtuelle y est notamment utilisée pour la réadaptation de patients paraplégiques [Kizony et al. 2002], traiter les désordres de déficit d'attention et d'hyperactivité, traiter les désordres de stress post-traumatique [Rothbaum et al. 1999], faciliter la compréhension de la schizophrénie [Ku et al. 2003], traiter les personnes atteintes de phobies telles que la peur en avion [Wiederhold et al. 2002], la peur des hauteurs, l'arachnophobie [Carlin et al. 1997] et la claustrophobie.

Le simulateur développé dans le cadre de cette étude fait partie de la cinquième catégorie. Plusieurs travaux de recherche et produits sont d'ailleurs mis de l'avant pour doter la communauté médicale d'outils efficaces pour l'apprentissage et l'entraînement des médecins. Parmi les plus connus et les plus pertinents, par rapport à nos travaux, nous retrouvons (figure 2.4) :

- Le Virtual Objective Structured Clinical Exam (VOSCE) [Johnsen et al. 2006] : il s'agit d'un prototype permettant aux étudiants en médecine d'expérimenter l'interaction patient-médecin. Ce système semi-immersif et monoscopique, qui affiche un patient virtuel interactif, utilise un projecteur, un micro pour la reconnaissance vocale, un assistant numérique personnel et des caméras pour le repérage de la tête et de la main ;
- Le SIMENDO [Verdaasdonk et al. 2006] : un simulateur pour endoscopie en chirurgie générale, gynécologie, urologie et orthopédie. Ce simulateur, qui est devenu un produit commercial distribué par la compagnie DelltaTech, fonctionne avec un ordinateur portable et utilise deux systèmes haptiques afin de simuler un retour d'effort lors de l'intervention virtuelle ;
- Le Virtual Reality Minimally Invasive Surgery Trainer (MIST-VR) [Sutton et al. 1997] : un système d'entraînement et d'évaluation pour la chirurgie laparoscopique minimalement invasive. Il utilise un ordinateur avec écran conventionnel et deux instruments laparoscopiques équipés de senseurs de position afin de simuler des tâches chirurgicales spécifiques ;
- Un outil d'entraînement chirurgical pour l'arthroscopie du genou [Megali et al. 2002]. Ce système utilise un ordinateur avec écran conventionnel, un genou synthétique et des outils chirurgicaux équipés de dispositifs de repérage ;
- Le Computer Enhanced Laparoscopic Training System (CELTS) [Stylopoulos et al. 2003] : un système d'entraînement à la chirurgie laparoscopique qui utilise un

ordinateur avec écran conventionnel, des instruments laparoscopiques équipés de dispositifs de repérage et différents modèles synthétiques représentant les zones à opérer.



**Figure 2.4** : Outils d'apprentissage et d'entraînement médicaux connus : 1- VOSCE ; 2- SIMENDO (DeltaTech) ; 3- MIST-VR ; 4- Outil d'entraînement chirurgical pour l'arthroscopie du genou ; 5- Computer Enhanced Laparoscopic Training System (CELTS).

Remarquons que cette liste omet volontairement certains outils et simulateurs médicaux qui offrent en plus des fonctionnalités collaboratives. Ceux-ci seront présentés dans la section 2.2.3. La plupart des systèmes présentés précédemment se définissent comme



des outils en réalité virtuelle. Il est par contre possible de remarquer, par le type d'équipements utilisés et le type d'images projetées, que la nature de ces systèmes ne correspond pas à définition de la RV donnée précédemment. Plus précisément, la plupart de ces systèmes ne procurent pas d'immersion dans la simulation (ils utilisent des écrans conventionnels), n'affichent pas d'images stéréoscopiques et n'offrent pas à l'utilisateur une interface lui permettant d'interagir directement et intuitivement avec les objets virtuels. Il existe donc une dissociation entre les gestes que pose l'utilisateur (par exemple, avec les outils chirurgicaux et les modèles synthétiques) et la représentation virtuelle de ces objets. Plusieurs travaux de recherche et produits commerciaux abusent d'ailleurs de ce terme et qualifient des simulateurs et jeux vidéos, fonctionnant avec une interface clavier-souris standard, d'environnements en réalité virtuelle. La « vraie » réalité virtuelle offre néanmoins un net avantage, par rapport à ces interfaces conventionnelles, qui justifie son utilité pour l'entraînement chirurgical [Plouznikoff 2004] :

« Bien qu'une interface proposant un pointeur souris semble à première vue plus facile, précise et intuitive à manipuler qu'une interface en réalité virtuelle, en pratique, tous les écrans d'ordinateurs présentent des marques de doigts. Cela prouve bien que les utilisateurs tentent de transcender de telles interfaces qu'ils jugent non intuitives. En immergeant l'utilisateur au sein d'un monde quasi-réel, la réalité virtuelle peut proposer une interface manuelle directe où des gestes courants telle la saisie d'objets sont reconnus sans besoin de faire appel à des menus ou des boutons, réduisant ainsi la courbe d'apprentissage du logiciel, permettant à l'utilisateur de se concentrer uniquement sur la tâche à accomplir. L'immersion complète permet de surcroît d'amplifier le réalisme du simulateur en proposant une reconstruction complète tridimensionnelle du patient, de la colonne et d'une salle d'opération virtuelle déjà connue, augmentant de ce fait le sentiment d'appartenance de l'utilisateur dans ce monde virtuel. Finalement, en combinant le monde réel et la scène opératoire, la réalité virtuelle permet d'augmenter la coordination main oeil comparativement aux interfaces standards où les deux sont largement dissociés (Poston and Serra 1996) ».

Malgré cet avantage majeur, il n'existe actuellement aucun simulateur d'entraînement chirurgical pouvant prétendre recréer l'expérience réelle d'une chirurgie à l'aide d'environnements de réalité virtuelle tels que la CAVE ou l'Immersadesk (à l'exception

de VRS3). Plusieurs outils médicaux à vocation plus générale sont par contre disponibles. Voici certains d'entre eux :

- Un système, conçu pour l'Immersadesk, permettant d'explorer et d'interagir avec des données volumétriques représentant l'anatomie virtuelle du corps humain [Lin et al. 2002] ;
- Le Virtual Temporal Bone : un environnement télé-immersif pour enseigner l'anatomie complexe de l'oreille interne et moyenne à l'aide d'un Immersadesk [Rasmussen et al. 1998] ;
- Un système, conçu pour l'Immersadesk, simulant la détérioration visuelle causée par certaines maladies des yeux telles que le glaucome, la dégénérescence maculaire liée à l'âge et la diplopie [Ai et al. 2000].

Il n'existe actuellement aucun outil préopératoire ou d'entraînement, utilisant une modélisation biomécanique simulant la colonne vertébrale scoliotique et son instrumentation, qui propose une salle d'opération virtuelle au sein de laquelle un chirurgien orthopédiste peut tester différentes stratégies opératoires pour un patient spécifique [Plouznikoff 2004]. Ce constat et l'exploration de la littérature, effectués pour cette étude, soutiennent le fait qu'il n'existe aucune version collaborative permettant à plusieurs chirurgiens de coopérer lors de sessions d'entraînement ou de télé-enseignement. Les travaux se rapprochant le plus de VRS3 et CVrS3 proviennent de Nigel et de ses collaborateurs (de l'Université de Manchester en Angleterre) : un simulateur non spécifique d'insertion de vis pédiculaires sur écran conventionnel.

Au cours des dernières années, plusieurs études ont été entreprises dans le but de déterminer l'utilité de la réalité virtuelle pour l'entraînement en chirurgie. Parmi celles-ci, l'étude de Grantcharov et Seymour a montré que l'utilisation préalable de MIST-VR, avant une vraie chirurgie laparoscopique, améliore la performance des chirurgiens lors de l'exécution de tâches spécifiques [Grantcharov et al. 2004] [Seymour et al. 2002].

Plus spécifiquement et plus récemment, une étude de Blyth et de ses collaborateurs s'est donnée l'objectif de déterminer l'opinion des chirurgiens orthopédistes à propos des simulateurs de chirurgie orthopédique en réalité virtuelle [Blyth et al. 2006]. Cette étude a révélé que 68% des chirurgiens croient qu'il est important de pratiquer dans un environnement simulé, que 75% croient que la simulation est un moyen efficace d'améliorer le retour d'information par rapport aux performances et que 77% croient qu'un simulateur peut être utile pour la planification préopératoire. Il est important de mentionner que cette évaluation ne précise pas le type de technologie qui caractérise les simulateurs. Aucune distinction n'est faite entre les simulateurs avec écran conventionnel, semi-immersif ou immersif, entre ceux offrant une vue 3D monoscopique ou stéréoscopique, entre ceux utilisant une interface avec souris-clavier ou ceux utilisant des dispositifs pour repérer les gestes de l'utilisateur. Les statistiques suivantes (provenant de la même étude) sont par contre très encourageantes quant à l'utilité de la « vraie » réalité virtuelle : 90% des chirurgiens croient qu'une vue réaliste de l'opération est nécessaire et 85% croient qu'une vue 3D doit être générée.

## **2.2 Environnements virtuels de collaboration**

Avant de présenter quelques-uns des environnements virtuels de collaboration (EVC) médicaux connus, une description succincte du domaine des EVC et de ses concepts incontournables est proposée. Les domaines d'application et des exemples d'EVC divers et éducatifs sont également présentés.

### **2.2.1 Les EVC : définition et caractéristiques**

Un environnement virtuel de collaboration consiste en un monde virtuel où plusieurs utilisateurs, provenant généralement d'endroits géographiquement éloignés, peuvent se voir, communiquer et interagir entre eux dans l'objectif de collaborer à une ou plusieurs tâches de nature spécifique ou générale. Les EVC sont généralement des environnements de réalité virtuelle mis en réseau afin de permettre la transmission des

moindres actions d'un utilisateur à tous les autres (déplacement, interaction avec des objets, ...), une communication textuelle et/ou vocale, le partage de données de travail et la transmission de vidéo capturant les visages des usagers afin de les projeter sur leurs avatars respectifs. La communication vocale et la transmission de vidéo ne sont par contre pas toujours utilisées puisqu'elles ne sont pas indispensables à tous les contextes collaboratifs et qu'elles sont gourmandes en terme de bande passante.

Les caractéristiques suivantes sont considérées, par Meiguins [Meiguins et al. 2004] et plusieurs autres chercheurs, comme étant à la base de tout EVC :

- **Une impression d'espace partagé** : les utilisateurs doivent avoir l'impression d'être au même endroit. Le lieu virtuel doit présenter une scène commune quasi-identique pour tous les utilisateurs ;
- **Une impression de présence partagée** : les utilisateurs doivent ressentir la présence des autres utilisateurs et pouvoir interagir entre eux. Les avatars avec ou sans vidéo sont normalement utilisés afin de procurer ce sentiment ;
- **Une impression de temps commun** : la plupart des actions et des conséquences aux actions d'un utilisateur doivent être perçues au même moment par tous les utilisateurs. Certaines actions nécessitent une propagation s'approchant du temps réel (déplacement d'un usager, interaction entre usagers, ...) tandis que ce n'est pas toujours indispensable pour d'autres ;
- **Une manière de communiquer** : un EVC doit offrir un mécanisme de communication entre usagers pour s'assurer qu'une collaboration est possible. Il peut s'agir de communication textuelle, vocale et/ou gestuelle ;
- **Une manière de partager** : les aspects précédents peuvent caractériser un système avancé de vidéoconférence. Le vrai pouvoir d'un CVE n'est donc pas seulement de permettre une interaction entre usagers, mais de pouvoir aussi interagir avec l'environnement en soit.

## L'architecture réseautique d'un EVC

La nature distribuée d'un EVC implique que son architecture réseautique est d'une grande importance quant à sa performance, sa facilité d'utilisation et aux possibilités qu'il peut offrir. Une revue succincte des concepts d'architecture réseautique est donc proposée.

Les données transmises sont généralement catégorisées (tableau 2.1) selon leur type et leur taille.

**Tableau 2.1** : Types de données transmises dans un EVC.

Type de données	Taille	Description
Événements	Petite	Il s'agit de données, telles que les positions d'un utilisateur, qui ne nécessitent pas une transmission fiable <sup>8</sup> ou de données notifiant un changement d'état qui, au contraire, nécessitent une transmission avec acheminement assuré.
Atomiques	Moyenne	Il s'agit de données, telle que la géométrie d'un objet, devant être traitées de façon atomique (comme un tout). Elles sont généralement assez petites pour pouvoir demeurer en mémoire physique du client local.
Segmentées	Grande	Il s'agit de données trop grosses pour résider entièrement en mémoire (telles que certaines données scientifiques). Elles sont donc accédées de façon segmentée.

Un EVC peut adopter différentes stratégies de diffusion, ou des combinaisons de celles-ci, pour l'échange des données entre ses entités<sup>9</sup>. Le tableau suivant présente les types de diffusion possibles.

<sup>8</sup> Assurant leur acheminement au destinataire.

<sup>9</sup> Clients et/ou serveurs.

**Tableau 2.2** : Types de diffusions réseautiques.

Type de diffusion	Description
Individuelle ( <i>unicast</i> )	Les messages sont envoyés directement d'un client à un autre. Il s'agit du type de diffusion le plus portable et le mieux adapté à une interconnexion pair-à-pair.
Générale ( <i>broadcast</i> )	Les messages sont envoyés à tous les autres clients du sous-réseau. Ce type de diffusion est simple à implanter, mais les messages ne traversent pas le sous-réseau à moins que les routeurs ne le fassent explicitement.
Multidiffusion ( <i>multicast</i> )	Les messages sont envoyés à un ensemble de clients spécifiés. Ce type de diffusion est très efficace et utilise la bande passante de manière optimale. Il requiert du matériel réseau le supportant.

La topologie réseau d'un EVC influence grandement sa performance. Le choix d'une topologie particulière dépend largement du domaine d'application et de l'usage planifié. Le tableau 2.3 décrit les topologies les plus couramment utilisées.

**Tableau 2.3** : Types de topologies réseautique couramment utilisés dans les EVC.

Type de topologie	Description
Répliquée - homogène	Chaque client possède sa propre copie complètement répliquée de l'état du monde virtuel. La cohérence entre les versions est assurée par la multidiffusion de messages à tous les autres clients lors du changement d'état d'une copie locale. Cette topologie est typique des EVC militaires.
Centralisée - partagée	Les données partagées sont conservées sur un serveur central qui gère de manière stricte leur cohérence entre les divers clients. Le fait que les messages transmis doivent parcourir le chemin en double peut affecter la latence du système. Un bris ou des problèmes du serveur impliquent l'impossibilité ou la difficulté à utiliser l'application.

Distribuée – pair-à-pair	Une structure de mémoire partagée est simulée sur le réseau virtuel créé par l'interconnexion des clients entre eux. Lorsqu'un client crée un objet, celui-ci est immédiatement répliqué chez les autres. Le développement d'un EVC est simplifié, mais la performance peut être affectée puisque chaque client doit établir une connexion avec tous les autres. Sa gestion devient de plus en plus difficile et de moins en moins performante à mesure que le nombre d'utilisateurs augmente.
Centralisée avec regroupements	Les données du monde virtuel sont réparties sur plusieurs serveurs interconnectés et chaque client choisit son serveur selon des critères d'accessibilité et de performance, ou selon le sous-monde désiré (géré par un serveur spécifique). Plusieurs jeux vidéo massivement multijoueurs utilisent cette topologie.

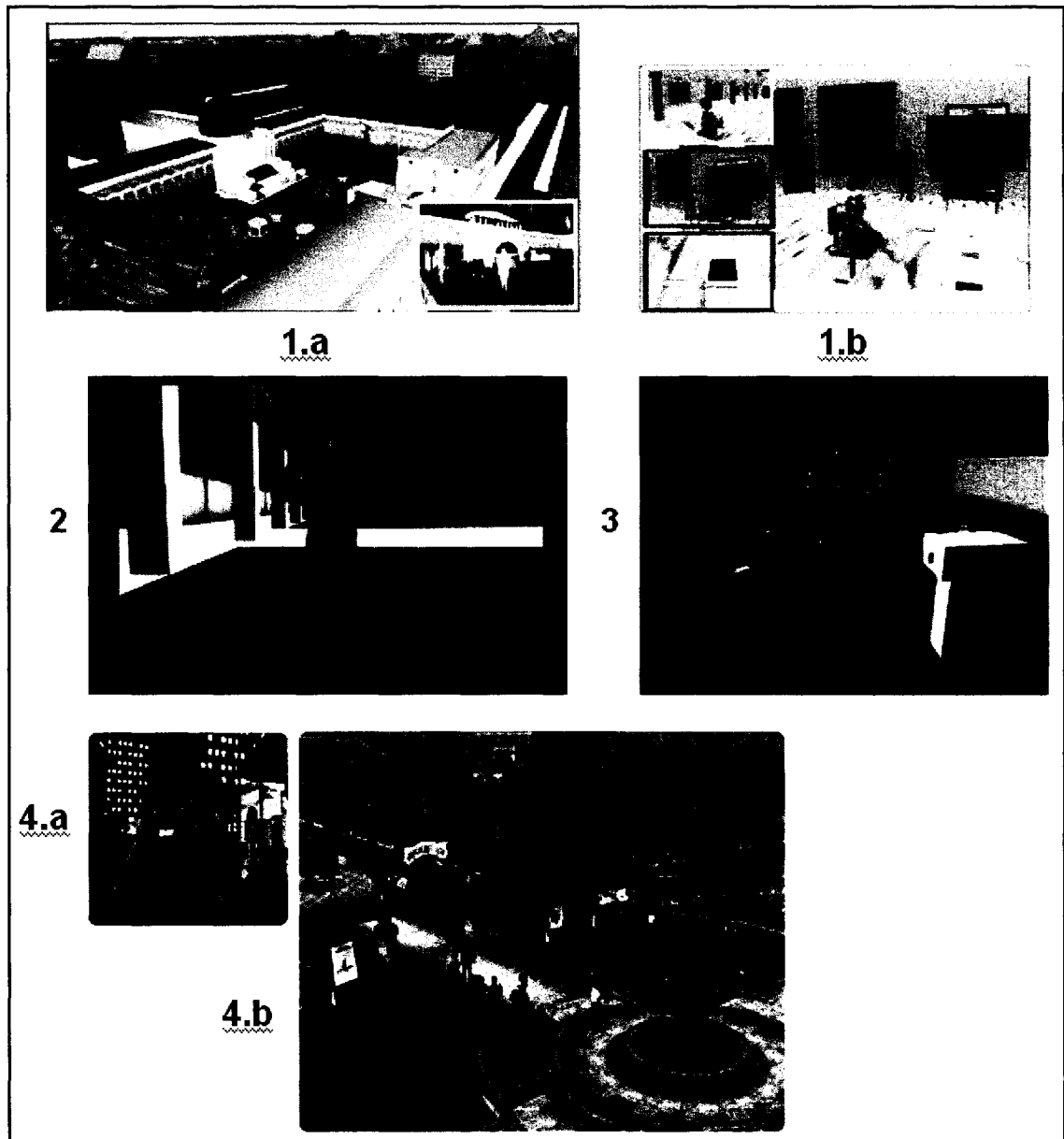
### 2.2.2 Domaines d'application des EVC

Les EVC peuvent être séparés en deux grandes catégories : les EVC à grande échelle (figure 2.5) et les EVC spécialisés. Les EVC à grande échelle se définissent comme des environnements adaptables pouvant potentiellement supporter tous types de collaboration. Ils sont conçus de façon à permettre aux usagers la création d'applications<sup>10</sup> spécifiquement désignées pour fonctionner à l'intérieur du monde virtuel de l'EVC. Cette extensibilité est la raison pour laquelle ce type d'environnement peut hypothétiquement supporter tous types de tâches collaboratives : lorsque l'EVC ne permet pas d'effectuer une tâche collaborative spécifique, il suffit de créer une application destinée à cette fin. Un EVC à grande échelle peut donc être perçu comme un monde virtuel (généralement semblable à une ville constituée d'immeubles, de parcs et de routes, mais qui offre aussi des zones non calquées sur le réel) ou un ensemble de mondes virtuels interconnectés à l'intérieur desquels il est possible de se déplacer, de

---

<sup>10</sup> Généralement à l'aide de scripts relativement faciles à générer.

communiquer entre utilisateurs et de, par exemple, tenir des conférences, éditer des géométries 3D, créer de nouvelles zones, jouer à des jeux, ...



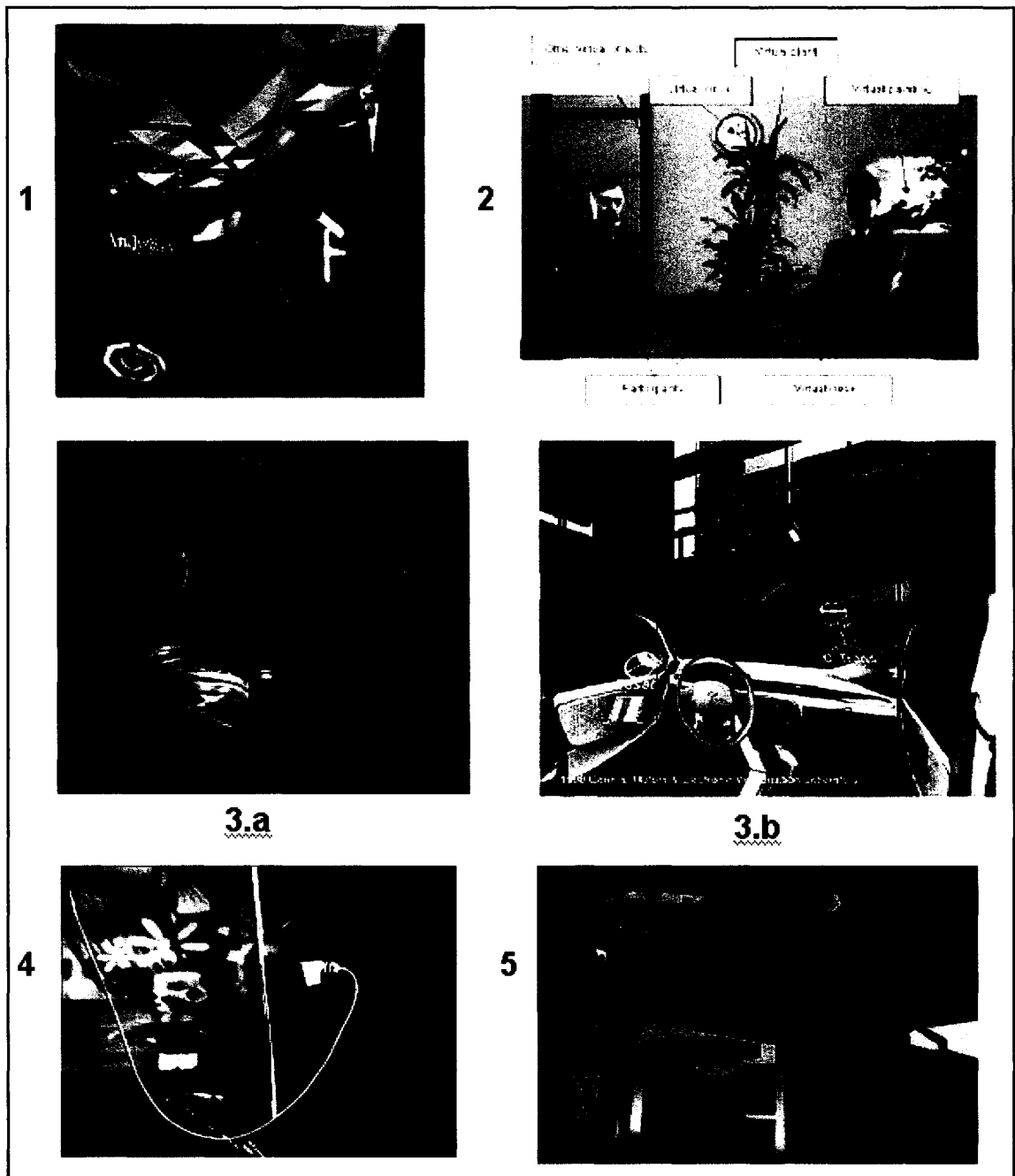
**Figure 2.5 :** EVC à grande échelle connus : 1- a et b DIVE (Distributed Interactive Virtual Environment) ; 2- NPSNET-IV ; 3- MASSIVE (Model, Architecture and System for Spatial Interaction in Virtual Environments) ; 4- a et b Second Life.

La philosophie régissant ce type de CVE est ambitieuse et intéressante, mais il faut par contre remarquer qu'il ne s'agit pas d'une plateforme idéale pour créer des applications



collaboratives spécialisées et complexes qui nécessitent une gestion réseautique stricte, une propagation de certaines actions s'approchant du temps réel, une grande variété d'interactions, l'intégration de modules spécialisés et/ou l'utilisation d'équipements sophistiqués (tels que des systèmes haptiques et des dispositifs de réalité virtuelle).

Les EVC spécialisés sont utilisés dans plusieurs domaines tels que la visualisation scientifique, la médecine, l'ingénierie, l'architecture, le design et le militaire. Ils sont conçus afin de permettre une collaboration entre plusieurs utilisateurs lors de l'exécution de tâches spécialisées (figure 2.6) tels que la revue de design d'automobile [Smith 2001], la tenue de session de vidéoconférence immersive [Kauff et Schreer 2002], l'entraînement militaire, visualiser et tester l'impact de la chute de produits [Wu et al. 1998], l'éducation des enfants, ... Ils sont normalement optimisés pour répondre aux besoins spécifiques de la tâche collaborative concernée.



**Figure 2.6 :** Quelques EVC spécialisés connus : 1- LIMBO, système de visualisation scientifique ; 2- I3DVC (Immersive 3D Videoconferencing), système de vidéoconférence 3D immersive ; 3- a et b VisualEyes (General Motors), outil de revue de design d'automobile ; 4- NICE (Narrative Immersive Constructionist/Collaborative Environments), CVE pour éduquer les enfants ; 5- CALVIN (Collaborative Architectural Layout Via Immersive Navigation), outil de design intérieur et d'architecture.

### **2.2.3 Les applications collaboratives et EVC médicaux**

Il existe une grande variété d'applications collaboratives conçues pour le domaine médical. Leur complexité, les technologies utilisées et leurs objectifs varient aussi énormément.

Certaines ne consistent qu'en des références hypertextes pouvant être éditées de manière collaborative. Le projet CHORUS [Kahn 2006], qui offre une base de données de références radiologiques pour les médecins et les étudiants en médecine, est l'un d'entre eux. Il permet notamment la soumission d'articles de nature médicale et la revue par les pairs via un site web conçu à cet effet.

Les systèmes médicaux de support de décision en groupe méritent d'être mentionnés. Le système de support collaboratif pour la prise de décision médicale de Rao et Turoff [Rao et Turoff 2000] est l'un d'entre eux. Il offre une interface collaborative qui permet la spécification, la configuration et l'organisation des éléments et des paramètres logiquement impliquées lors de l'établissement de diagnostics médicaux. Mentionnons qu'un système expert est également incorporé afin d'aider à la prise de décision et de proposer des pistes de solutions. Le système Asthma Kiosk [Porter et al. 2004], une technologie centrée sur le patient qui permet la prise de décision collaborative au département d'urgence, fait également partie de cette catégorie.

Les prochaines sous-sections décrivent les applications collaboratives et les EVC, spécialisés pour l'enseignement médical et la simulation chirurgicale, les plus connus et les plus pertinents par rapport à nos travaux.

### **Un système semi-immersif et collaboratif pour l'éducation médicale**

Choi et ses collaborateurs travaillent sur une application collaborative d'éducation médicale avancée qui, à l'aide d'un ou plusieurs Virtual Workbench<sup>11</sup>, permet une manipulation interactive des tissus mous et simule dynamiquement le comportement des organes humains grâce à une modélisation déformable de ceux-ci [Choi et al. 2005]. Ce projet permet à plusieurs utilisateurs, assis autour de l'écran, d'observer simultanément le comportement d'organes du corps humain, mais il ne peut être contrôlé que par une seule personne. Le fait qu'il ne permette qu'une collaboration locale (ses utilisateurs doivent être au même endroit) justifie, avant toutes autres considérations, pourquoi il ne peut être considéré comme un EVC.

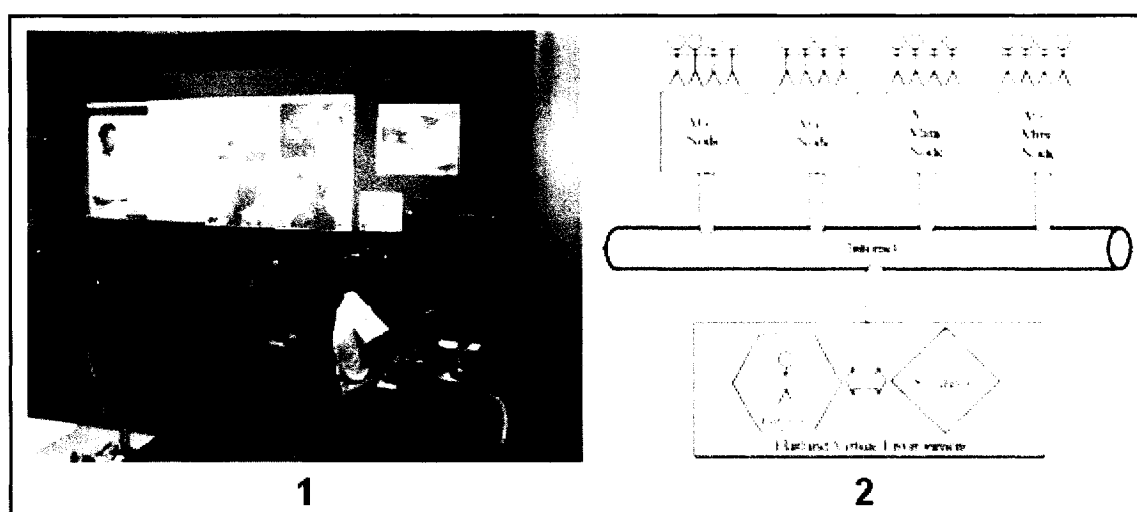
### **Le projet TOUCH**

L'objectif du projet TOUCH (Telehealth Outreach for Unified Community Health) [Caudell et al. 2003] est de développer un environnement informatique permettant l'apprentissage dirigé en groupe. Le groupe est constitué d'individus situés à des endroits différents qui effectuent en coopération des exercices en rapport au traitement d'un patient virtuel, avec intelligence artificielle, qui simule les symptômes de problèmes médicaux spécifiques. Chaque groupe est situé dans un studio équipé de projecteurs, de caméras et de micros afin que tous les membres puissent communiquer, se voir et visualiser les résultats de chaque intervention. Le fait qu'un seul des studios (figure 2.7.2) soit équipé d'un visiocasque, d'une souris 3D et de systèmes de repérage pour permettre à un étudiant d'intervenir sur le patient virtuel est la raison pour laquelle TOUCH ne peut être considéré comme un EVC. Une impression d'espace et de temps partagés n'est pas générée puisqu'un seul des participants est représenté et peut interagir à l'intérieur du monde virtuel. Les moindres actions de l'étudiant sont par contre transmises à tous les autres studios où il est possible de choisir une vue appropriée sur la

---

<sup>11</sup> Système de réalité virtuelle semi-immersif avec affichage stéréoscopique, semblable à l'Immersadesk, qui est orienté de façon à être parallèle au sol (comme pour une table).

scène. Chaque studio (ou nœud AG) utilise le système National Computational Science Alliance Access Grid (NCSA/AG) afin de gérer une session de vidéoconférence en permettant une collaboration et une communication temps réel, multipoint et groupe à groupe. Ce système utilise une topologie réseau « centralisée - partagée » qui conserve, adapte et simule l'état du patient, par rapport aux requêtes lui étant envoyées, sur un seul serveur (figure 2.7.1) : celui avec lequel l'utilisateur équipé du visiocasque interagit avec le patient.



**Figure 2.7 :** Le projet TOUCH : 1- Le studio AG à partir duquel l'utilisateur équipé d'un visiocasque interagit avec le patient virtuel ; 2- Architecture réseau globale de TOUCH [Caudell et al. 2003].

### **Une application d'entraînement télé-chirurgical de la cataracte dans un environnement de collaboration haptique et visuel**

El-Far et ses collaborateurs travaillent sur une application visuelle et haptique pouvant être utilisée par des étudiants en médecine pour s'entraîner à la chirurgie de la cataracte [El-Far et al. 2005]. Il s'agit d'une application conçue pour être utilisée avec un ordinateur avec écran conventionnel, un système haptique et, un micro et une caméra pour supporter la vidéoconférence. Elle permet notamment qu'un instructeur supervise et guide à distance les manoeuvres des étudiants, lors de l'entraînement, à l'aide de systèmes haptiques présents à tous les endroits. Un participant n'est pas contraint d'utiliser un système haptique pour participer à la session : il peut être passif et jouer le

rôle d'observateur. L'architecture réseau de ce système (figure 2.8) nécessite une communication directe entre les modules de contrôle des systèmes haptiques de chaque client. L'objectif est d'assurer qu'une synchronisation haptique s'approchant du temps réel, à l'aide d'algorithmes de prédiction physique, est possible lorsqu'un instructeur guide les mouvements des autres participants. Le fait que l'application utilise le réseau photonique CANARIE<sup>12</sup> pour les communications réseaux participe également à la fluidité de son fonctionnement. La topologie globale du système est de type « distribuée – pair-à-pair » et l'état visuel de l'œil du patient est conservé en mémoire par tous les participants. Les fonctionnalités collaboratives telles que la vidéoconférence sont assurées par un intergiciel (« middleware ») qui est connecté à tous les clients. Cette application peut être considérée comme un EVC dans le sens où elle satisfait les cinq caractéristiques de Meiguins présentées dans la section 2.2.1. Il faut par contre remarquer qu'elle procure une faible impression de présence partagée puisque seuls les outils des participants sont représentés dans la scène virtuelle. Cette faiblesse est compensée par l'utilisation d'une fenêtre graphique parallèle de vidéoconférence. Une telle dissociation est malgré tout non souhaitable en rapport à la philosophie générale des EVC où tout doit être intégré à l'intérieur d'une scène commune quasi-identique.

---

<sup>12</sup> Réseau canadien optique à très haut débit réunissant les réseaux scientifiques provinciaux et, par leur entremise, les universités, les instituts de recherche et les laboratoires publics.

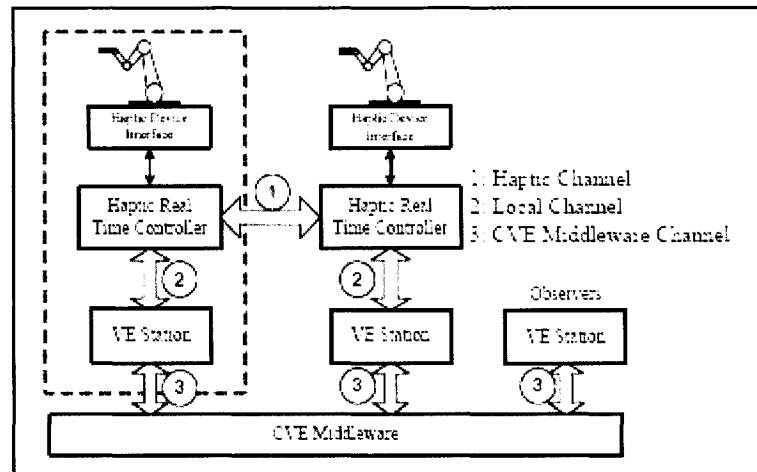


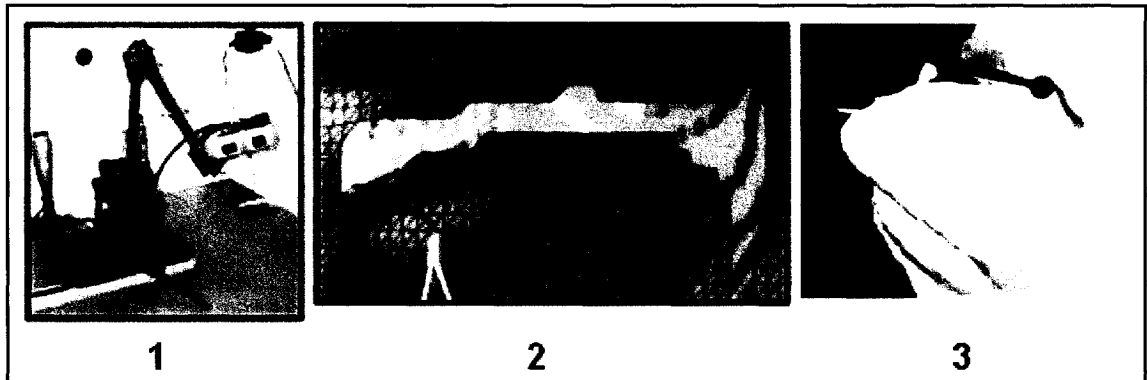
Figure 2.8 : Architecture réseau du simulateur collaboratif de chirurgie de la cataracte.

### Un EVC pour la simulation de la chirurgie de l'os temporal

L'EVC développé par Morris et ses collaborateurs permet de s'entraîner de manière coopérative à la chirurgie de l'os temporal à l'aide de dispositifs binoculaires stéréoscopiques<sup>13</sup> et de systèmes haptiques [Morris et al. 2004]. La dissection de l'os est simulée visuellement (figure 2.9) et haptiquement à l'aide d'une représentation hybride des données qui permet de maintenir des surfaces lisses pour l'affichage graphique et d'utiliser des données volumétriques pour la rétroaction haptique. Il offre une rétroaction auditive à l'aide de sons de perceuse, basés sur des données expérimentales, et il simule une console de surveillance de nerfs virtuels. Le système permet aux utilisateurs indépendants d'observer et de manipuler un modèle commun et il permet à un utilisateur de ressentir la force générée par le contact de l'os avec la perceuse contrôlée par un autre utilisateur. Ce système utilise une topologie réseautique de type « répliquée-homogène » où chaque client possède une copie du modèle de l'os et où seulement les modifications du modèle et les forces/positions de la perceuse sont transmises via le réseau. Il satisfait finalement les cinq caractéristiques de Meiguins d'un EVC, mais il procure une faible

<sup>13</sup> Une lunette formée de deux petits écrans LCD montée sur un bras ajustable. Ce dispositif est semblable aux microscopes binoculaires utilisés lors d'une vraie chirurgie de l'os temporal.

impression de présence partagée puisque seuls les outils des participants sont représentés dans la scène virtuelle.



**Figure 2.9** : Simulateur collaboratif de chirurgie de l'os temporal : 1- Dispositif binoculaire utilisé ; 2- Dissection de la peau recouvrant l'os ; 3- Perçage virtuel de la surface de l'os temporal.

Malgré l'augmentation de l'intérêt de la communauté scientifique face à l'utilisation de la réalité virtuelle et des EVC pour l'entraînement et l'enseignement médical, une faible utilisation en contexte clinique peut être observée. Mantovani identifie à ce propos plusieurs problèmes devant être attaqués afin de faciliter l'intégration de la RV et des EVC pour l'entraînement et l'enseignement [Mantovani 2003] :

- **Les coûts** : le coût des environnements de réalité virtuelle telle que la CAVE est très élevé ;
- **Le manque de standards de référence** : la plupart des applications créées dans ce domaine sont « uniques » dans le sens où elles sont dépendantes de dispositifs et de logiciels propriétaires qui ont été adaptés selon un processus d'essais et d'erreurs. Elles sont donc difficilement adaptables à des contextes différents de ceux dans lesquels elles ont été développées ;
- **La culture éducative** : les technologies graphiques 3D ne sont pas conçues pour remplacer entièrement les techniques d'enseignement conventionnelles. Ces technologies doivent être intégrées et gérées par les enseignants dans le contexte éducationnel actuel. Les enseignants doivent aussi acquérir l'expertise spécifique



et une expérience pratique suffisante afin de pouvoir supporter un processus d'enseignement efficace via les environnements 3D d'enseignement ;

- **La sécurité** : certains utilisateurs éprouvent des effets secondaires durant et après l'exposition aux environnements de RV immersifs (problèmes oculaires, désorientation, nausée, etc.). Malgré le fait que les outils de RV les plus récents semblent causer moins ou pas d'effets secondaires, il s'agit d'un aspect important à surveiller ;
- **Utilisabilité** : les systèmes de RV d'entraînement et d'enseignement doivent être développés via la participation des utilisateurs potentiels. Cette tâche difficile nécessite un design collaboratif et itératif du système qui doit impliquer, aux différentes étapes du projet, les développeurs, les enseignants et les utilisateurs potentiels.

La présente recherche propose un EVC spécialisé conçu pour la simulation de la chirurgie scoliotique. Le système élaboré peut être comparé aux simulateurs collaboratifs de chirurgie de la cataracte et de l'os temporal présentés précédemment, mais attribue une emphase différente aux caractéristiques de Meiguins. En offrant une vue représentant l'ensemble de la salle de chirurgie, plutôt qu'une vue limitée à la zone opératoire, il a pour objectif d'accroître le sentiment d'immersion et d'espace partagé. L'utilisation d'avatars, qui représentent les corps des participants, et non seulement leurs outils, a pour objectif d'augmenter l'impression de présence partagée.

Mentionnons finalement que le choix de créer une version du simulateur pour interface conventionnelle, avec écran, souris et clavier standards, se veut une réponse au problème du coût des environnements de RV avancés mentionné par Mantovani.

Le prochain chapitre entame la description de l'architecture du simulateur développé et présente les choix d'implémentation majeurs à la base du prototype résultant.

## **CHAPITRE 3 : ÉLABORATION DE L'ARCHITECTURE DU SIMULATEUR**

Ce chapitre présente l'architecture globale de CVrS3 qui est constituée de trois entités principales : le serveur biomécanique d'instrumentation collaborative du rachis scoliotique, le serveur de téléprésence multi-usagers et le client de simulation de chirurgie scoliotique en réalité virtuelle. Les arguments ayant menés au choix d'une telle architecture sont présentés ainsi qu'un survol succinct des rôles de chaque entité. Les outils et les données initiales utilisés pour l'élaboration du simulateur sont également présentés. Chaque entité sera présentée plus en détail dans les trois prochains chapitres.

### **3.1 Architecture proposée**

La figure 3.1 montre l'architecture globale du système qui met en lumière l'organisation et les liens entre les différentes entités. Le matériel utilisé pour soutenir une session collaborative typique est représenté. Chaque entité est décomposée de manière abstraite en ses composantes ou modules principaux. Cette décomposition offre une vue extrêmement simplifiée des entités concernées, mais permet d'obtenir une meilleure compréhension des principales tâches y étant accomplies.

Le système développé utilise deux serveurs dédiés et mutuellement indépendants pour la gestion de l'instrumentation collaborative de la colonne vertébrale scoliotique et pour la gestion de la téléprésence entre les utilisateurs. Ce choix est justifié par le fait que ces tâches sont facilement dissociables et qu'ils n'ont en aucun cas besoin d'accéder à des informations mutuelles. La seule exception concerne les informations d'identification des clients qui participent à la session : elles sont donc répliquées sur les deux entités.

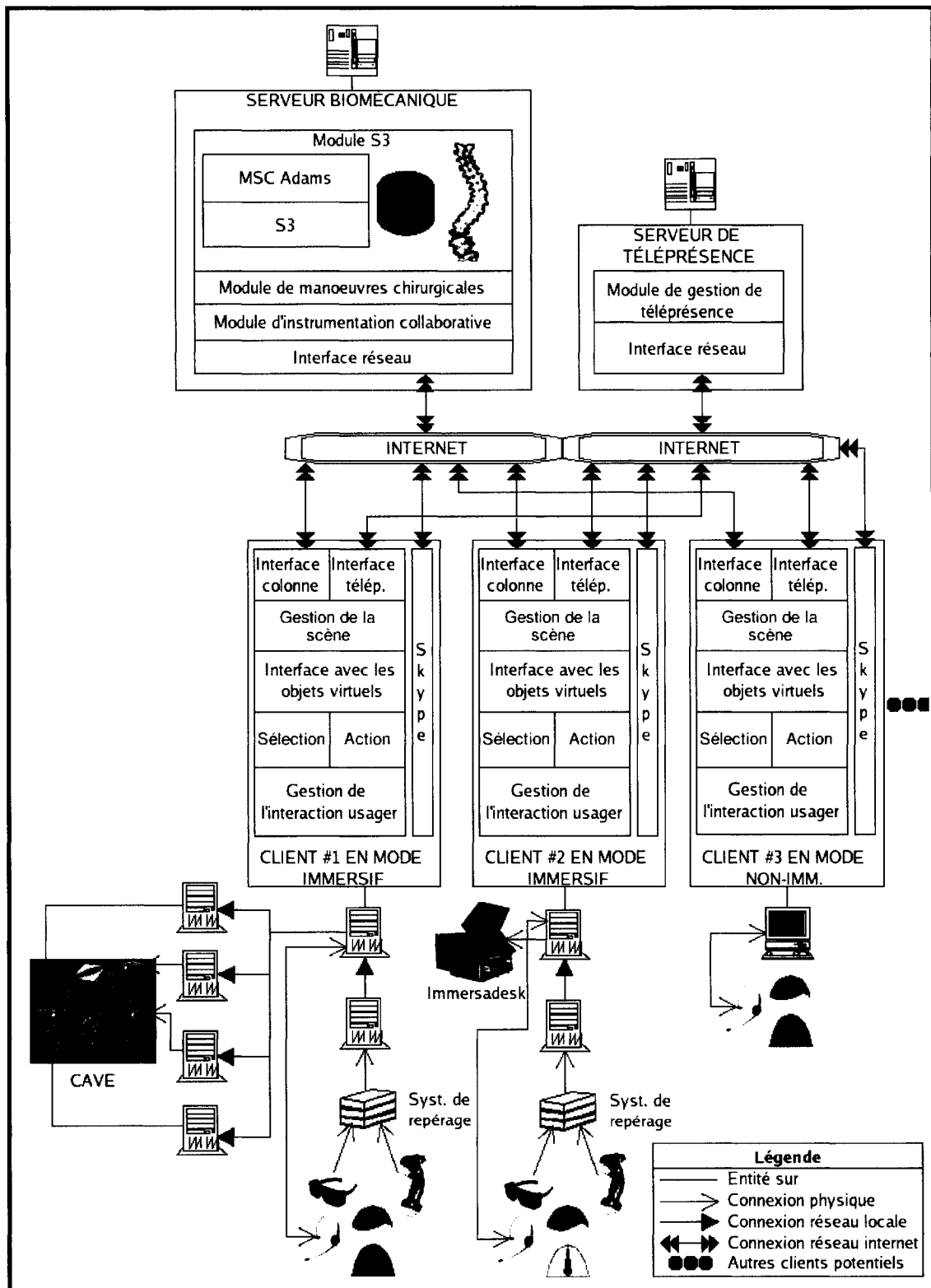


Figure 3.1 : Architecture globale de CVrS3.

Cette séparation est également justifiée par la volonté d'éviter le couplage entre les entités. L'objectif est de faciliter l'intégration future de nouvelles versions du moteur biomécanique ou la création de nouveaux serveurs biomécaniques conçus pour d'autres types de chirurgie orthopédique. Le fait que le serveur biomécanique actuel nécessite une très forte puissance de calcul a également motivé son isolement. Le but est d'assurer qu'il puisse répondre aux requêtes dans un délai minimal. Le serveur de téléprésence peut, du même coup, propager ses données avec une latence minimale.

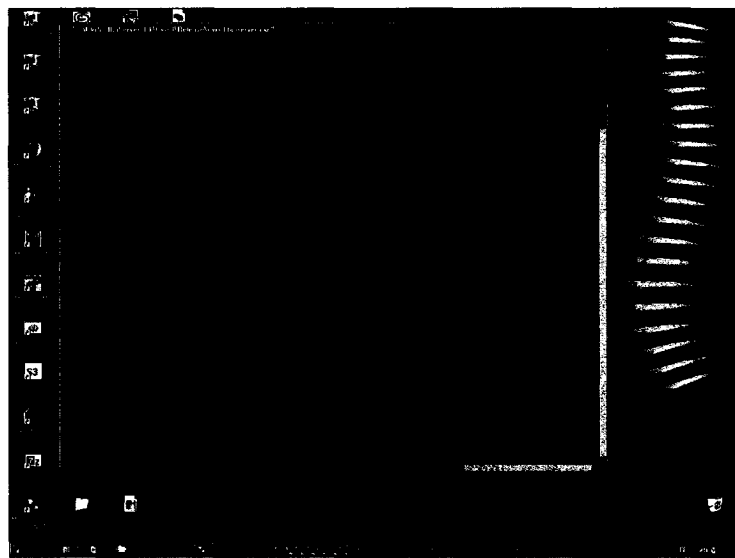
### **3.1.1 Serveur biomécanique d'instrumentation collaborative du rachis scoliotique**

La topologie réseau choisie pour la gestion de l'état de la colonne est de type « centralisée – partagée » où les données sont conservées sur un serveur central qui gère de manière stricte leur cohérence entre les divers clients. Utiliser le moteur biomécanique sur chaque client afin d'avoir plusieurs copies aurait, en plus de rendre les clients moins performants, complexifié la gestion des situations où deux ou plusieurs utilisateurs effectuent des requêtes différentes et quasi-simultanées sur leur propre copie. Des algorithmes de contrôle de concurrence auraient pu être implémentés pour régler ce problème, mais ils auraient provoqué une complexification des communications réseaux : les clients auraient dû continuellement s'envoyer des messages de contrôle. Plus le nombre de clients serait élevé et plus cette situation serait difficile à gérer.

Le serveur biomécanique accepte, lors de l'initialisation d'une session d'entraînement, les connexions des clients. Son rôle est d'accepter et d'appliquer les requêtes pour ensuite propager les réponses à tous les autres clients. Les requêtes correspondent à des demandes d'insertion, de changement de position et de suppression d'implants, à des demandes d'attachement, détachement de tige métallique préalablement courbée et à des demandes d'application de manœuvres spécifiques de correction. Les réponses consistent principalement en des matrices de position et d'orientation des implants et des vertèbres, et aux positions des points de contrôle des tiges. La réponse à la requête

spécifique d'un client est multidiffusée à tous les autres afin qu'ils puissent travailler sur une seule et même représentation de la colonne.

La figure 3.2 montre la console du serveur qui affiche l'état de la colonne, les requêtes des utilisateurs et les réponses conséquentes lors d'une session d'instrumentation. Plus d'informations sur le fonctionnement du serveur et sur les modules et composants le constituant seront présentées dans le quatrième chapitre.



**Figure 3.2 :** Console du serveur biomécanique.

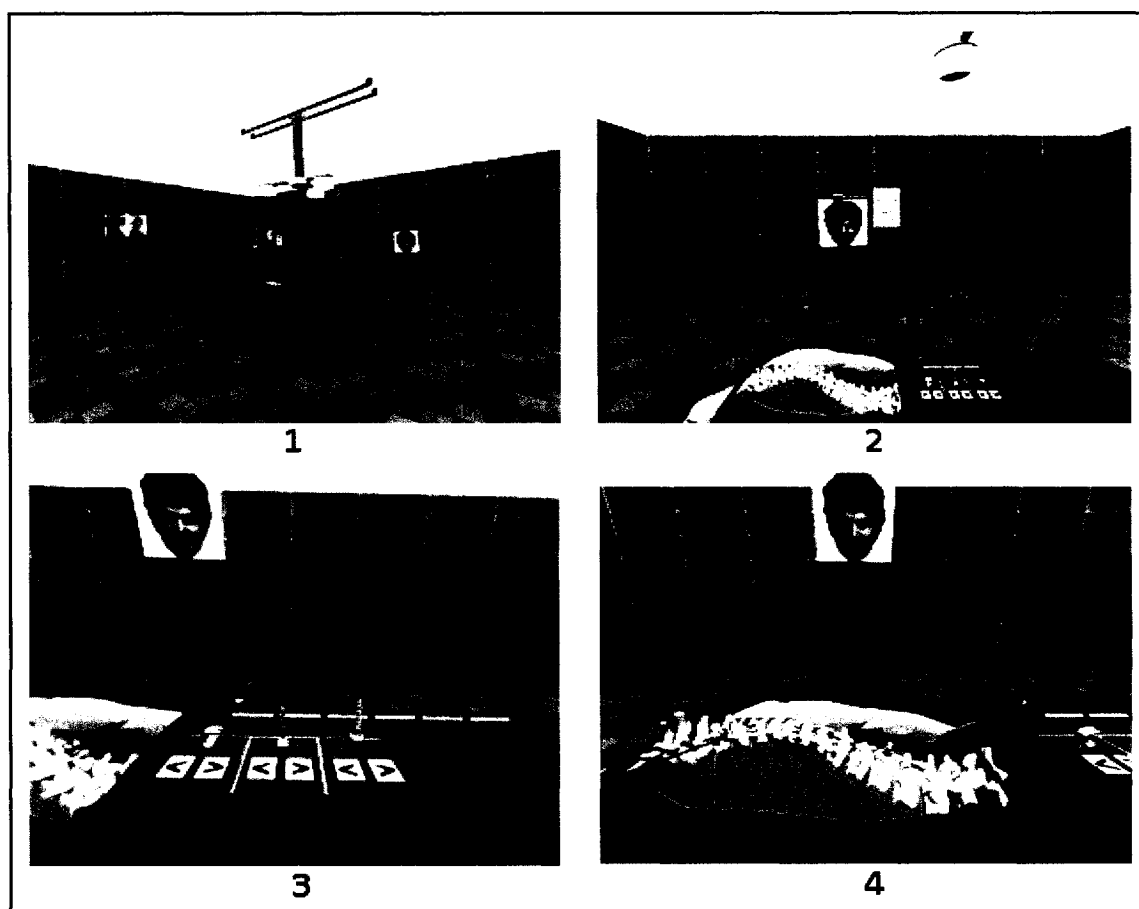
### **3.1.2 Serveur de téléprésence multi-usagers**

Le serveur de téléprésence utilise lui aussi une topologie de type « centralisée – partagée ». Son rôle est de propager, sous la forme d'un flux continu, les matrices de position et d'orientation de la tête et de la souris 3D, d'un utilisateur en mouvement, à tous les autres. Le résultat est que chaque utilisateur peut voir les déplacements des avatars et des souris 3D des autres utilisateurs. La topologie choisie a permis de faciliter le développement en évitant les problèmes de gestion réseautique qu'aurait provoqué une communication pair-à-pair : gestion de la connexion/déconnexion des clients sur chaque client, congestion du réseau causée par son inondation lorsqu'un nombre élevé d'utilisateurs participent à la session, ... Le fait que les informations transmises doivent

parcourir le double du chemin pourrait provoquer un problème de latence, mais cet aspect est jugé négligeable étant donné le haut débit du réseau à fibres optiques de l'École Polytechnique qui est utilisé pour le développement du prototype. Tous les tests usagers ont d'ailleurs été effectués via ce même réseau. Plus d'informations à propos de ce serveur seront présentées dans le cinquième chapitre.

### **3.1.3 Client de simulation de chirurgie scoliotique en réalité virtuelle**

Le client est l'application que doit lancer un utilisateur afin de participer à une session d'entraînement à l'instrumentation du rachis scoliotique en réalité virtuelle. Il s'agit d'une application infographique 3D qui gère l'affichage et permet à l'utilisateur d'interagir avec les objets d'une salle de chirurgie virtuelle. La scène (figure 3.3) est principalement constituée d'une lampe chirurgicale, des radiographies du patient, d'étagères, d'une table d'opération, d'un drap chirurgical, d'un plateau avec implants et outils, de manettes de sélection d'implants, du corps d'un patient et de la colonne vertébrale du patient à opérer.



**Figure 3.3** : Captures d'écran du client CVrS3 en mode non-immersif<sup>14</sup> : 1- Vue globale de la scène ;  
2- Vue rapprochée du patient ; 3- Vue rapprochée du plateau à implants/outils ;  
4- Vue rapprochée du rachis scoliotique.

Le client est conçu de façon à pouvoir être utilisé en mode immersif ou non-immersif : avec un environnement de réalité virtuelle (tel que la CAVE et l'Immersadesk) ou avec un ordinateur avec écran conventionnel, clavier et souris. En mode immersif (figure 3.4), le repérage des lunettes (qui donne la position/orientation de la tête) permet à l'utilisateur de se déplacer légèrement et de s'orienter à l'intérieur de la salle. La vue

<sup>14</sup> Le mode non-immersif du client utilise le mode simulation de VR Juggler (outil avec lequel est développé CVrS3) qui permet de développer et de tester un logiciel de réalité virtuelle à partir d'un ordinateur conventionnel sans avoir à utiliser un dispositif de RV spécialisé. L'objet vert est un artéfact représentant la souris 3D de l'utilisateur.

stéréoscopique de la scène, adaptée continuellement à sa perspective, l'immerge dans une salle de chirurgie virtuelle très réaliste dont l'expérience est fort semblable à celle d'une vraie salle. L'interaction avec les objets se fait via la souris 3D. Celle-ci est également connectée au système de repérage afin que sa position 3D soit toujours connue du logiciel. Elle peut être considérée comme une extension de la main avec laquelle l'on peut sélectionner et déplacer des implants, courber une tige chirurgicale, insérer des implants dans les vertèbres, attacher une tige aux implants, effectuer des manœuvres correctrices, ...



**Figure 3.4 :** CVrS3 en mode immersif<sup>15</sup> (CAVE de Polytechnique).

---

<sup>15</sup> L'affichage de la CAVE est en mode monoscopique afin de rendre la photo plus claire.



Le mode non-immersif (figure 3.5) procure une vue 3D sur la scène semblable à celle qu'on retrouve dans les jeux vidéos. Le fait que cette vue 3D soit simulée sur un petit écran 2D, non stéréoscopique, et qu'il faille utiliser le clavier et la souris pour interagir avec les objets nuit grandement au sentiment d'immersion et au réalisme de la simulation. Néanmoins, cette possibilité rend l'utilisation du simulateur beaucoup plus accessible étant donné la rareté et le coût des environnements de réalité virtuelle tels que la CAVE. Le mode non-immersif est calqué sur le mode immersif : plutôt que de se déplacer physiquement et d'utiliser la vraie souris 3D, l'utilisateur se déplace virtuellement à l'aide du clavier et doit contrôler le mouvement d'une représentation de la souris 3D à l'aide de la souris standard. Étant donné la nature spatiale des actions à effectuer lors d'une chirurgie, ce type de contrôle rend l'instrumentation très difficile. L'avantage est qu'il est malgré tout possible d'observer le comportement des autres utilisateurs et de communiquer avec eux. Pour un contexte de télé-enseignement, cette version non-immersive s'avère donc fort utile.



**Figure 3.5 :** CVrS3 en mode non-immersif.

Une session d'entraînement débute à partir du moment où un premier client est lancé. L'un des premiers processus du client est de se connecter au serveur biomécanique afin de l'initialiser et d'obtenir les données qui caractérisent l'état de la colonne. Une connexion avec le serveur de téléprésence s'effectue parallèlement afin d'obtenir, s'ils existent, les informations des autres clients. Une session d'instrumentation individuelle peut malgré tout avoir lieu si aucun autre client n'est connecté.

Les clients qui se connectent subséquemment effectuent le même processus. Pour chaque nouveau client connecté au serveur de téléprésence, un nouvel avatar fait son apparition dans la scène de tous les autres. L'action d'un utilisateur sur la colonne (insertion d'implant, attachement de tige, ...), implique l'envoi d'une requête au serveur biomécanique. Ce dernier applique la requête et transmet la réponse à tous les autres clients afin qu'ils puissent être témoins de l'action de leur confrère.

## **3.2 Outils et données initiales**

Cette section présente les outils et les données qui étaient accessibles lors du développement du prototype.

### **3.2.1 CAVE et Immersadesk**

La CAVE de l'École Polytechnique et l'Immersadesk du LMBCAO ont été utilisés pour développer et tester le logiciel en mode immersif. La CAVE est actuellement configurée de façon à fonctionner avec une grappe de PC Linux. Quatre ordinateurs, équipés de cartes graphiques spécialisés, sont dédiés à la gestion de l'affichage sur les quatre écrans de la voûte d'immersion. Un cinquième ordinateur, à partir duquel le logiciel est lancé, joue le rôle de maître et s'occupe de la gestion et de la synchronisation des écrans. La sixième machine est dédiée à la gestion du dispositif de repérage qui retourne les positions de la souris 3D et des lunettes. L'Immersadesk du LMBCAO n'utilise que deux machines. L'une des machines est utilisée pour l'affichage tandis que l'autre gère le système de repérage. Les applications de réalité virtuelle développées pour la CAVE

de Polytechnique et l'Immersadesk du LMBCAO doivent être basées sur VR Juggler. Ce dernier est présenté dans la prochaine sous-section.

### **3.2.2 VR Juggler et OpenSG**

VR Juggler est une collection de technologies qui procurent les outils nécessaires pour le développement d'applications de réalité virtuelle. Cette application à source ouverte (« open source ») fournit des bibliothèques multi-plateformes, basées C++, qui permettent l'orchestration des différents dispositifs pouvant être utilisés pour la création de logiciel de réalité virtuelle : système de repérage, souris 3D, gants de RV, visiocasque, écran avec positionnement particulier, ... VR Juggler est conçu de façon à pouvoir être facilement utilisable avec une CAVE gérée par une grappe de PC.

OpenSG est un gestionnaire de graphe de scène en C++ qui facilite le développement d'applications infographiques en offrant des fonctionnalités fort utiles telles que l'importation de modèles géométriques, la détection de collisions, des bibliothèques mathématiques, ... OpenSG est multi-plateforme, à source ouverte et il est basé sur OpenGL. Il est conçu de façon à pouvoir être intégré facilement avec VR Juggler.

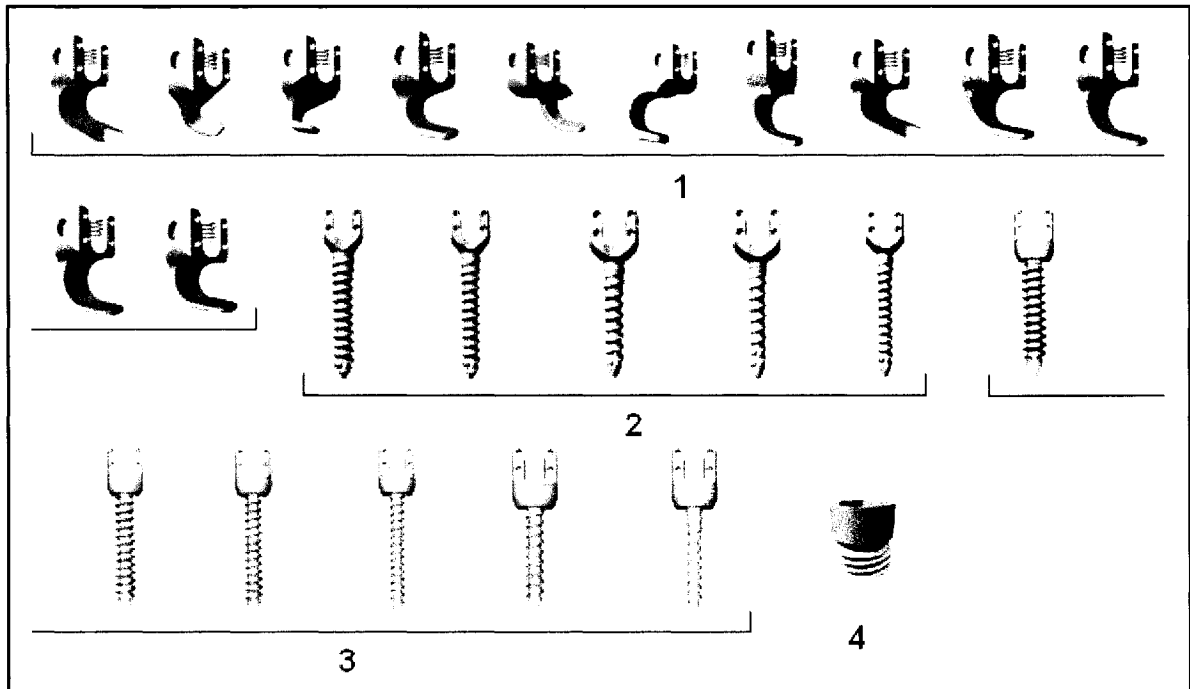
OpenSG conserve l'ensemble de la scène dans un graphe organisé de manière hiérarchique. L'objectif est d'optimiser la façon dont les données sont transmises au matériel graphique. Le fait qu'il vérifie la visibilité des éléments de la scène, par rapport au point de vue de l'utilisateur, permet d'éliminer les parties cachées du processus d'affichage. Un gain des performances d'affichage est donc obtenu. Un graphe OpenSG est typiquement peuplé de nœuds interconnectés. Chaque nœud contient un seul type de données. Les types de données suivants peuvent être utilisés : géométries, transformations, matériels (textures, couleurs, ...), éléments d'aiguillage (« switches »), lumières, etc.

### 3.2.3 C++ Sockets Library

C++ Sockets Library est une librairie C++ multi-plateforme à source ouverte qui englobe l'API (Application Programming Interface) des sockets de Berkeley. Elle est conçue de façon à faciliter le développement d'application réseau en offrant un cadre d'interface simplifié qui interagit avec les fonctions de base de gestion de connexion, d'acceptation, d'écriture et de lecture via les sockets. Les modules de communication réseau présents dans le client de simulation chirurgicale, le serveur biomécanique et le serveur de téléprésence utilisent cette librairie.

### 3.2.4 Spine Surgery Simulator

Le code source de S3, l'outil de planification préopératoire pour la chirurgie scoliotique développé au LMBCAO, a été fourni. Le serveur d'instrumentation collaborative développé pour cette étude intègre une partie de son code. Seules les parties du code qui s'occupent de la modélisation biomécanique du rachis ont été incorporées. Les autres parties de S3 ont été omises puisqu'ils s'occupaient principalement de la gestion du système de fenêtrage. L'un des objectifs de CVrS3 est justement de proposer une alternative au type d'interaction procuré par les fenêtres en les remplaçant par une interface en réalité virtuelle. Les modèles géométriques VRML des implants, utilisés pour la simulation de l'instrumentation dans S3, ont également été incorporés (figure 3.6). Le modèle biomécanique de S3, qui régit le comportement de la colonne en fonction des forces correctrices appliquées, est conçu avec le SDK Adams de MSC Software. Plus d'informations à propos du modèle seront présentées dans le prochain chapitre.



**Figure 3.6** : Modèles géométriques des implants de S3 (CD Horizon, Medtronic) : 1- Crochets ; 2- Vis pédiculaires fixes ; 3- Vis pédiculaires multiaxiales ; 4- Écrou.

### 3.2.5 Virtual Spine Surgery Simulator

Le code source de VRS3, le prédécesseur de CVrS3, a également été fourni. Bien qu'aucune partie de son code n'ait été utilisée, il a constitué une source d'inspiration quant à certaines de ses fonctionnalités et à l'apparence globale de la salle de chirurgie. Les modèles géométriques des armoires, de la table d'opération et du drap chirurgical ont été incorporés dans CVrS3.

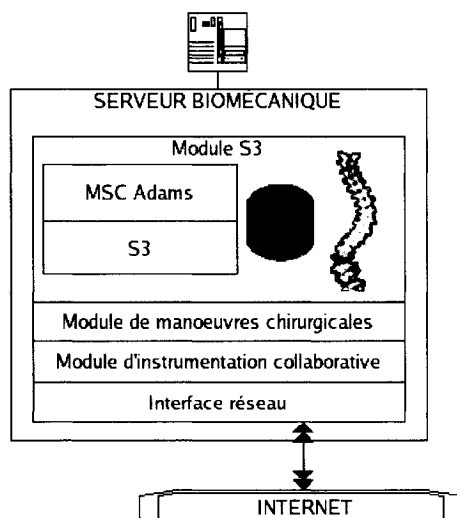
## **CHAPITRE 4 : SERVEUR BIOMÉCANIQUE D'INSTRUMENTATION COLLABORATIVE DU RACHIS SCOLIOTIQUE**

L'objectif du présent chapitre est de décrire, d'une manière plus approfondie, les composants du serveur biomécanique développés pour gérer l'instrumentation collaborative du rachis scoliotique. Les sections qui suivent présentent le serveur dans son ensemble et procurent par la suite une description détaillée du module S3, du module de manœuvres chirurgicales et de l'interface réseau.

### **4.1 Description globale du serveur développé**

#### **4.1.1 Détails d'implémentation et principaux composants**

L'intégration d'une partie du code de S3, conçu pour Windows, a forcé le développement d'un serveur conçu pour cette même plateforme. Le serveur est développé en C++ et il utilise la librairie C++ Sockets Library afin de gérer les communications réseaux avec les clients. L'ordinateur dédié pour son exécution est un Pentium III 996 MHz avec 512 méga-octets de mémoire vive. Afin de faciliter sa compréhension, le serveur peut être séparé en quatre composantes principales : le module S3, le module de manœuvres chirurgicales, le module d'instrumentation collaborative et l'interface réseau (figure 4.1).



**Figure 4.1 :** Composantes du serveur biomécanique d'instrumentation collaborative du rachis scoliotique.

Le module S3 correspond à la partie du code de S3 et au logiciel qui ont été intégrés pour gérer l'instrumentation du rachis scoliotique d'un patient spécifique. Les sections de code à vocations plus générales et celles qui gèrent l'interface utilisateur de S3 ont été omises lors de l'intégration. La boîte « S3 » représente le code écrit avec le SDK Adams qui modélise le comportement biomécanique de la colonne en fonction des forces correctrices lui étant appliquées via l'application de manoeuvres. Ce code interagit avec le logiciel de modélisation biomécanique MSC Adams qui est installé sur le serveur. Les données biomécaniques, qui résident dans la base de données de MSC Adams, correspondent, par exemple, aux points de contrôle des tiges attachées, aux matrices de position/orientation des implants insérés et des vertèbres.

Le module de manoeuvres chirurgicales est une interface vers le module S3. Il offre une série de fonctionnalités dédiées pour chaque type d'action et de manoeuvre chirurgicale qu'il est possible d'appliquer lors d'une session d'instrumentation.

L'objectif du module d'instrumentation collaborative est de gérer la concurrence et la priorité entre les requêtes d'instrumentation des clients participant à une même session. Il fonctionne actuellement selon le principe FIFO (First In, First Out) où les premières requêtes reçues sont traitées en premier. Une requête reçue lors de l'application d'une

autre doit donc attendre que celle-ci soit terminée avant de pouvoir être appliquée. Il répond également à un autre objectif : adapter les fonctionnalités du module de manœuvres chirurgicales pour une utilisation avec interface en réalité virtuelle.

L'interface réseau sert à accepter et gérer les connexions des clients qui participent à une session d'instrumentation. Elle s'occupe de recevoir, d'analyser et d'appliquer les requêtes d'actions spécifiques sur la colonne et de multidiffuser les réponses à tous les clients connectés. Elle s'assure également qu'un client, lors de sa connexion, reçoive les informations caractérisant l'état initial du rachis du patient. Le contenu des requêtes et des réponses conséquentes est exprimé sous la forme de chaîne de caractères. Le protocole TCP (Transmission Control Protocol) est utilisé pour les communications entre le serveur et les clients afin d'assurer une transmission fiable, avec connexion, des envois réseaux.

#### **4.1.2 Diagramme de classe du serveur**

Le diagramme de classe simplifié du serveur biomécanique est présenté dans l'annexe I. Le serveur est séparé en trois ensembles contenant respectivement les classes de gestion de l'interface réseau, de l'instrumentation collaborative en réalité virtuelle et de la modélisation biomécanique du rachis scoliotique. Les classes du premier ensemble proviennent de la librairie C++ Sockets Library et peuvent être associées à la boîte « Interface réseau » de la figure 4.1. Les classes du deuxième ensemble ont été créées afin de gérer l'instrumentation collaborative en réalité virtuelle. Deux d'entre elles correspondent au module d'instrumentation collaborative de la figure 4.1 tandis que la troisième joue le rôle du module de manœuvres chirurgicales. Les classes du troisième ensemble correspondent au module S3 de la figure 4.1. Il s'agit des classes du projet S3 qui ont été intégrées pour gérer la modélisation du rachis scoliotique d'un patient spécifique. Les détails et les explications concernant le diagramme de classe sont présentés dans l'annexe I.



## 4.2 Module S3

### 4.2.1 Reconstruction 3D du rachis d'un patient spécifique

Avant de pouvoir s'entraîner à l'instrumentation du rachis scoliotique d'un patient spécifique, il faut appliquer une méthode de reconstruction 3D afin d'obtenir une description tridimensionnelle et biomécanique de la colonne vertébrale du patient. Le logiciel Spine3D (figure 4.2), développé au Laboratoire Informatique de Scoliose 3D (LIS3D) et au LMBCAO, applique une méthode qui utilise comme données entrantes les radiographies numériques, prises selon deux points de vue (PA et LAT<sup>16</sup>), de la colonne du patient concerné. Un gilet de calibrage est d'abord porté par le patient lors d'une prise de radiographies en salle de radiologie. Les radiographies sont ensuite traitées par un outil d'analyse numérique afin d'identifier les repères anatomiques qui devront être reconstruits en 3D. Une vingtaine de repères anatomiques par vertèbre sont typiquement identifiés pour effectuer la reconstruction 3D standard des vertèbres. Les données 2D recueillies, qui correspondent aux positions de chaque repère sur les images radiographiques, sont enregistrées dans un fichier.

Les coordonnées 2D du fichier et les coordonnées 3D qui caractérisent l'objet de calibrage sont ensuite utilisées pour définir les équations de reconstruction 3D. La reconstruction autocalibrée de tous les points correspondants des vertèbres et des points non correspondants des plateaux vertébraux est ensuite effectuée. Le fichier qui résulte de ce processus porte l'extension ".o3". Il s'agit du fichier qu'il faut spécifier au serveur afin d'initialiser le module S3.

---

<sup>16</sup> Vue antéropostérieure et latérale.



Figure 4.2 : Capture d'écran du logiciel Spine 3D.

#### 4.2.2 Modèle biomécanique du rachis

Le modèle biomécanique [Luce 2004] [Aubin et al. 2006, 2], développé avec le SDK Adams, utilise une approche de modélisation cinétique basée sur les équations différentielles d'équations dynamiques et algébriques qui décrivent les contraintes du système sous la forme générale suivante :

$$M\ddot{q} + Kq + C^T \lambda = Q_e + Q_r$$

où  $M$  est la matrice de masse des corps rigides,  $K$  est la matrice de rigidité des éléments flexibles,  $C$  sont les équations algébriques qui représentent les contraintes du système,  $Q_e$  et  $Q_r$  sont respectivement les forces externes et les forces de réaction,  $q$  est la matrice des coordonnées généralisées des éléments du système et  $\lambda$  est un vecteur des coefficients de Lagrange. Les équations incluent les propriétés mécaniques de la colonne, les contraintes du système d'instrumentation de la colonne et les charges externes, appliquées durant la chirurgie, combinées aux forces de réaction du système.

Le modèle est formé de corps rigides qui correspondent aux vertèbres thoraciques, lombaires et sacrées, et d'éléments flexibles qui représentent les structures

intervertébrales. Un système local (vertébral) de coordonnées est défini au centroïde de chaque corps vertébral. Leurs positions et orientations sont obtenues par la méthode de reconstruction 3D appliquée sur les radiographies du patient. Une représentation géométrique détaillée de chaque vertèbre est obtenue grâce à un atlas de vertèbres déformées par une technique de déformation en forme libre (krigeage dual) qui adapte les points reconstruits. Le comportement global des structures intervertébrales déformables est représenté par un élément (une matrice de rigidité 6x6) qui permet la spécification de relations non linéaires entre les déplacements (rotation et/ou translation), et les forces de réactions et moments pour six degrés de liberté. Les valeurs initiales des termes de la matrice de rigidité sont définies à partir d'approximations linéaires de données expérimentales in vitro provenant de la littérature [Panjabi et al. 1976]. Les valeurs de rigidité sont ensuite ajustées pour tenir compte de la distance intervertébrale relative, spécifique au patient, à l'aide de la méthode de Gardner-Morse et de ses collaborateurs [Gardner-Morse et al. 1990]. Finalement, les valeurs de rigidité sont précisées régionalement à l'aide de radiographies latérales de flexion du patient et d'un algorithme d'optimisation qui minimise la divergence entre la flexion simulée et la courbure mesurée des trois segments principaux de la colonne (courbes scoliotiques thoracique proximale, thoracique principale et thoraco-lombaire/lombaire) [Petit et al. 2004].

Les implants (crochets, vis fixes, les deux parties des vis multiaxiales) sont modélisés comme des corps rigides. Afin d'obtenir un comportement réaliste et adéquat de leurs connexions avec la colonne et les tiges, des contraintes généralisées de la forme  $\mathbf{C}^i(\mathbf{q}^i, t) = 0$  sont introduits pour restreindre les degrés de liberté appropriés et/ou représenter les propriétés de flexibilité appropriées.

Les joints (vis fixe)/vertèbre et crochet/vertèbre sont définis par des joints flexibles (« fields ») qui permettent une mobilité restreinte en rotation et en translation. Les coefficients de rigidité sont approximés à partir de données expérimentales prises sur des vertèbres cadavériques instrumentées. En ce qui concerne les vis multiaxiales, un joint

sphérique est défini entre la tête et le corps de la vis. Un ressort additionnel à trois degrés de liberté permet la spécification d'un comportement moment/angle. Avant le barrement de la vis, une grande rigidité est définie entre le corps de la vis et l'os (qui permet possiblement une légère déformation locale, mais une rigidité de rotation quasi-nulle est définie entre la tête et le corps de la vis jusqu'à ce que l'angle de la tête atteigne 27 degrés par rapport à l'axe de la vis. Lorsque cette rotation maximale est atteinte, une valeur de rigidité quasi-infinie est introduite pour simuler le blocage de la vis à la limite de son espace de déplacement. La tige de 5 mm est considérée flexible et possède les propriétés mécaniques de l'acier inoxydable ou du titane. Elle est représentée par la composition d'éléments de poutre déformable formulée selon la théorie de Timoshenko [Oden et Ripperger 1981].

Lors de l'exécution d'une manœuvre d'attachement de tige, des contraintes de déplacement, en translation et rotation, sont créées entre le segment de la tige et les implants ciblés pour simuler l'attachement. Des joints cylindriques sont ainsi introduits pour connecter les implants à la tige. Cette même procédure est répétée pour chaque tige additionnelle.

La manœuvre de rotation de tige implique l'application graduelle d'un couple sur la tige jusqu'à ce que l'angle désiré, par rapport à l'axe vertical global, soit atteint. Les implants sont libres de glisser et d'effectuer une rotation le long de la tige lors de l'application de cette manœuvre.

Le serrage d'écrou (sur un implant) est simulé par l'ajout d'une contrainte de déplacement qui bloque la translation le long de l'axe du joint cylindrique et la rotation autour de ce même joint qui correspond au lien entre l'implant et la tige. Pour les vis multiaxiales, des contraintes de déplacement sont également ajoutées pour bloquer la rotation de la tête de la vis par rapport à son corps.

La manœuvre de compression ou de distraction, qui permet de rapprocher ou d'éloigner deux implants le long de l'axe d'une tige, est simulée par l'application graduelle d'un

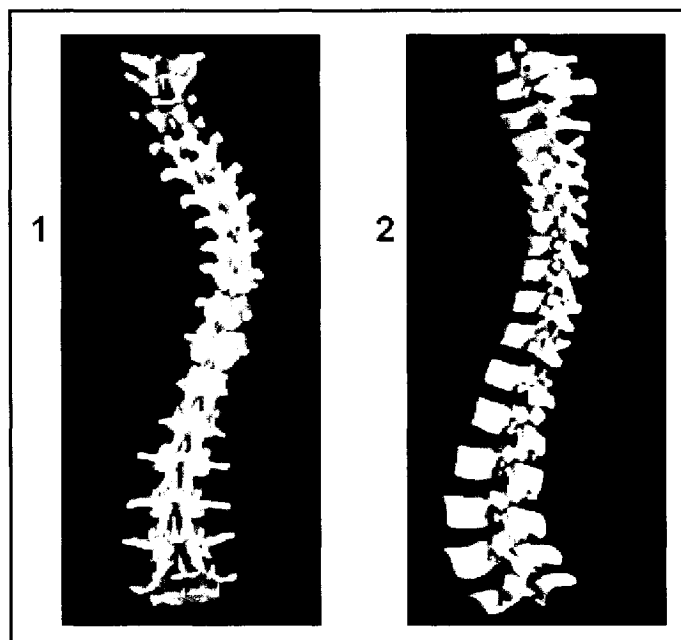
déplacement, sur les deux implants sélectionnés, jusqu'à ce que la distance inter-implant désirée soit atteinte.

La manœuvre de dérotation vertébrale directe (DVD) est simulée par l'application d'un couple sur la vertèbre sélectionnée jusqu'à ce que la rotation vertébrale désirée soit atteinte.

La solution non linéaire, subséquente à l'application de chacune des manœuvres mentionnées, est traitée grâce à la méthode de Newton-Raphson. Le critère de convergence est fixé à une force de déséquilibre maximale de 0.1 N, à un moment de déséquilibre maximal de 0.1 Nmm et à un déplacement relatif maximal, entre deux itérations, de 1 mm.

#### **4.2.3 Initialisation du module S3**

Le fichier ".o3" spécifique au patient (créé lors de l'étape de reconstruction 3D) est lu lors du lancement du serveur pour calculer les déformations géométriques permettant d'ajuster les données géométriques de référence (non déformées et provenant de mesures cadavériques) sur les données 3D personnalisées (provenant de la reconstruction 3D). Les déformations géométriques associées aux vertèbres sont calculées par krigeage dual avec une dérive et une covariance linéaires. Les paramètres de déformations géométriques enregistrés sont ensuite utilisés pour ajuster les données géométriques de référence sur les données de personnalisation. Cet ajustement qui est effectué sur un modèle nodal crée un fichier ".ng3" de nœuds géométriques déformés. Ce processus génère également un fichier VRML qui représente la géométrie 3D du rachis du patient concerné (figure 4.3).



**Figure 4.3** : Modèle VRML de la colonne vertébrale d'un patient : 1- Vue antéropostérieure ; 2- Vue latérale.

Le fichier ".ng3" est ensuite lu par le module S3 afin d'en obtenir une représentation en mémoire et de générer un modèle Adams. Toutes les données du fichier ".ng3" sont stockées dans l'arbre du modèle Adams qui résident à l'extérieur du module S3 dans la base de données de MSC Adams. Le module S3 utilise des méthodes d'objets *wrappers* pour accéder aux attributs afin d'éviter la duplication des données et les problèmes de synchronisation inhérents.

#### **4.2.4 Positions optimales d'implants chirurgicaux**

Le processus de reconstruction 3D et de modélisation géométrique fournit également une prévision des positions « optimales » des points d'insertion des vis pédiculaires et des crochets. Ils sont obtenus lors de la déformation par krigeage dual des vertèbres cadavériques par rapport aux dimensions spécifiques du patient. Les points, identifiés initialement sur les vertèbres cadavériques, sont transposés au cours de ce processus sur la reconstruction 3D de la colonne du patient.

Ces marqueurs sont utilisés par le simulateur afin de simplifier l'insertion des implants et la modification de leur position. Le tableau 4.1 présente les points d'insertion des vis et des crochets pour lesquels sont calculées les positions optimales. Un intervalle d'insertion vertébrale est associé à chaque position d'insertion pour chaque type d'implant.

**Tableau 4.1** : Positions des points d'insertion des implants.

Type d'implant	Position vertébrale générale	Position spécifique	Intervalle d'insertion
<b>Vis pédiculaire (fixe ou multiaxiale)</b>	Pédicule gauche ou droit	Standard	T1 – L5
		Anatomique	T1 – T12
<b>Crochet</b>	Pédicule gauche ou droit	Infra	T1 – T12
	Lamina gauche ou droit	Infra	T10 – L5
		Supra	T1 – L5
	Pédicule transverse gauche ou droit	Supra	T1 – T11

### 4.3 Module de manœuvres chirurgicales

La classe CAdamsManager (voir l'annexe I), qui correspond au module de manœuvres chirurgicales, offre plusieurs fonctionnalités. Ces fonctionnalités sont utilisées pour initialiser et gérer le moteur biomécanique, pour appliquer des manœuvres chirurgicales et pour obtenir de l'information quant à l'état de la colonne du patient en cours d'instrumentation. Plus précisément, des fonctions sont disponibles pour : l'insertion d'un implant à une position pré-calculée, la création d'une tige et la modélisation de sa courbure, l'insertion d'un implant à une position spécifique, l'enlèvement d'un implant, l'attachement d'une tige, le détachement d'une tige, la rotation d'une tige, le serrage d'une tige, la compression ou la distraction de vertèbres et la dérotation vertébrale directe (DVD). Une description des fonctions dédiées pour chaque manœuvre est présentée dans l'annexe II.

## **4.4 Interface réseau du serveur**

### **4.4.1 Processus de connexion et d'initialisation**

Le client est responsable d'initier une connexion avec le serveur lors de son lancement. Une requête d'initialisation du moteur biomécanique est envoyée par le client lorsque la connexion est établie. La requête n'est appliquée qu'une seule fois : elle est traitée lorsqu'un premier client la transmet, mais ignorée pour tous les autres. L'initialisation du serveur biomécanique implique les étapes suivantes :

1. Initialisation du logiciel MSC Adams et de sa base de données ;
2. Définition du jeu d'instruments (implants, outils, ...) utilisé pour l'instrumentation : CD Horizon de Medtronic ;
3. Lecture du fichier ".o3" qui caractérise la colonne du patient pour lequel la simulation est effectuée et génération des fichiers NG3 et VRML ;
4. Lecture du fichier ".ng3" et stockage des données dans la base de données de MSC Adams ;
5. Création et ouverture d'une session d'instrumentation du rachis scoliotique pour le patient spécifié ;
6. Obtention des matrices de position/orientation des vertèbres et conservation locale des données.

Lorsqu'un client a obtenu une confirmation de l'initialisation du serveur biomécanique :

1. Il envoie une requête d'obtention des marqueurs des points d'insertion pré-calculés. Le serveur traite la requête et lui transmet l'information demandée ;



2. Lorsque les marqueurs ont été reçus, ils envoient une requête d'obtention des matrices de position/orientation des vertèbres. Le serveur traite la requête et lui transmet l'information demandée.

La gestion de la cohérence de l'état de la colonne prend pour acquis une connexion préalable des clients. Les utilisateurs doivent donc s'assurer que tous les participants sont présents avant d'effectuer des manœuvres chirurgicales sur le rachis.

#### 4.4.2 Processus d'interaction pour l'instrumentation

L'interface réseau du serveur accepte plusieurs types de requête associés aux manœuvres d'instrumentation possibles. Le contenu de ces requêtes est formulé de façon à spécifier les paramètres nécessaires à l'application des manœuvres concernées. Les requêtes d'instrumentation qui sont gérées par l'interface réseau sont décrites dans le tableau 4.2.

**Tableau 4.2 :** Requêtes d'instrumentation gérées par l'interface réseau du serveur biomécanique.

<b>Requête :</b>	<b>Requête d'insertion d'un implant et d'obtention des données d'insertion.</b>
<b>Paramètres :</b>	Le numéro de la vertèbre d'insertion, la catégorie d'implant, le modèle de l'implant et la dernière position de l'implant avant l'insertion.
<b>Description :</b>	Le serveur trouve le point d'insertion pré-calculé le plus près de la position de l'implant spécifié. Il effectue ensuite une manœuvre d'insertion à partir du point trouvé et des autres paramètres. Lorsque l'application de la manœuvre est terminée et qu'elle a réussi, le serveur multidiffuse les données d'insertion à tous les clients connectés.
<b>Données retournées :</b>	Le numéro de la vertèbre, la catégorie de l'implant, le modèle de l'implant, le nouveau numéro d'identification de l'implant, le côté de la vertèbre implantée et la matrice de position/orientation de l'implant.

<b>Requête :</b>	<b>Requête de modification de position d'un implant vertébral et d'obtention des nouvelles données positionnelles.</b>
<b>Paramètres :</b>	Le numéro de la vertèbre de l'implant et le côté de la vertèbre implantée.
<b>Description :</b>	Le serveur identifie l'implant concerné à partir des paramètres et le déplace au prochain point d'insertion pré-calculé. Le changement de position est régi par les règles d'insertion du tableau 4.1. Lorsque l'application de la manœuvre est terminée et qu'elle a réussi, le serveur multidiffuse les nouvelles données d'insertion à tous les clients connectés. Un message d'échec est envoyé si le déplacement est impossible.
<b>Données retournées :</b>	En cas de réussite : le numéro de la vertèbre, le côté de la vertèbre implantée, le numéro d'identification de l'implant et la nouvelle matrice de position/orientation de l'implant. En cas d'échec : une chaîne de caractère signifiant l'échec de la requête.
<b>Requête :</b>	<b>Requête de suppression d'un implant vertébral et d'obtention d'une confirmation.</b>
<b>Paramètres :</b>	Le numéro de la vertèbre de l'implant et le côté de la vertèbre implantée.
<b>Description :</b>	Le serveur identifie l'implant concerné à partir des paramètres et le supprime. Lorsque la manœuvre a réussi, une confirmation de la suppression est multidiffusée à tous les clients.
<b>Données retournées :</b>	Une chaîne de caractère signifiant la réussite de la suppression, le numéro de la vertèbre de l'implant et le côté de la vertèbre qui était implantée.

<b>Requête :</b>	<b>Requête d'attachement complète d'une tige et d'obtention du nouvel état de la colonne.</b>
<b>Paramètres :</b>	L'ordre d'attachement, le côté vertébral des implants à attacher et les points de contrôle de la tige.
<b>Description :</b>	Le serveur crée une nouvelle tige à partir des points de contrôle et effectue une manœuvre d'attachement, selon l'ordre spécifié, avec tous les implants vertébraux situés sur le côté de la colonne spécifié. Lorsque l'application de la manœuvre est terminée et qu'elle a réussi, le serveur multidiffuse les données qui caractérisent le nouvel état de la colonne instrumentée. Un message d'échec est envoyé si le déplacement est impossible.
<b>Données retournées :</b>	En cas de réussite : les nouveaux points d'insertion pré-calculés, les nouvelles matrices de position/orientation des vertèbres, les numéros d'identification des tiges attachées et leurs nouveaux points de contrôle, les numéros d'identification des implants vertébraux et leurs nouvelles matrices de position/orientation. En cas d'échec : une chaîne de caractère signifiant l'échec de la requête.
<b>Requête :</b>	<b>Requête de détachement complet d'une tige et d'obtention du nouvel état de la colonne.</b>
<b>Paramètres :</b>	Le numéro d'identification de la tige.
<b>Description :</b>	Le serveur effectue une manœuvre de détachement complet de la tige. Lorsque l'application de la manœuvre est terminée et qu'elle a réussi, une confirmation du détachement pour la tige spécifiée est multidiffusée à tous les clients. Un message d'échec est envoyé si le détachement est impossible.
<b>Données retournées :</b>	En cas de réussite : une chaîne de caractère signifiant la réussite du détachement et le numéro d'identification de la tige détachée. En cas d'échec : une chaîne de caractère signifiant l'échec de la requête.

<b>Requête :</b>	<b>Requête de rotation de tige et d'obtention du nouvel état de la colonne.</b>
<b>Paramètres :</b>	Le numéro d'identification de la tige et l'angle de rotation désiré.
<b>Description :</b>	Le serveur effectue une manœuvre de rotation selon l'angle spécifié pour la tige. Lorsque l'application de la manœuvre est terminée et qu'elle a réussi, le serveur multidiffuse les données qui caractérisent le nouvel état de la colonne instrumentée. Un message d'échec est envoyé si la rotation est impossible.
<b>Données retournées :</b>	En cas de réussite : les nouveaux points d'insertion pré-calculés, les nouvelles matrices de position/orientation des vertèbres, les numéros d'identification des tiges attachées et leurs nouveaux points de contrôle, les numéros d'identification des implants vertébraux et leurs nouvelles matrices de position/orientation. En cas d'échec : une chaîne de caractère signifiant l'échec de la requête.

Le fait que certaines données des requêtes sont répétées dans les réponses pourrait, à première vue, être considéré comme un gaspillage de bande passante. Cette duplication est par contre voulue : l'objectif est d'assurer que les clients, n'étant pas au courant d'une requête envoyée par un autre client, puissent modifier l'état de la colonne en ayant seulement accès aux éléments de la réponse. La multidiffusion de réponses détaillées permet donc d'éviter la transmission de messages supplémentaires qui décriraient les intentions du client concerné.

#### 4.4.3 Format et structure des messages

Les données échangées via les requêtes et réponses sont toutes exprimées par une chaîne de caractères d'une seule ligne. Elles terminent toujours par un caractère de saut de ligne. Ce format d'échange a été choisi pour les messages afin de profiter des fonctionnalités offertes par la librairie C++ Sockets Library qui simplifient la gestion des transmissions réseaux. La plupart des messages réseaux utilisent une structure globale semblable à celle-ci :

ÉTIQUETTE Type\_de\_données<sub>0</sub>:données<sub>0</sub> Type\_de\_données<sub>1</sub>:données<sub>1</sub> ...  
 Type\_de\_données<sub>n</sub>:données<sub>n</sub>

où "ÉTIQUETTE" indique la nature de la requête ou de la réponse, "Type\_de\_données<sub>i</sub>:" indique le i<sup>e</sup> type des données qui suivront et "données<sub>i</sub>" représente les i<sup>e</sup> données du type déclaré. Un seul caractère d'espace doit exister entre "ÉTIQUETTE" et "Type\_de\_données<sub>0</sub>". Aucun espace ne doit séparer "Type\_de\_données<sub>i</sub>:" et "données<sub>i</sub>". Un seul caractère d'espace doit séparer les couples "Type\_de\_données<sub>i</sub>:données<sub>i</sub>" et "Type\_de\_données<sub>i+1</sub>:données<sub>i+1</sub>". Finalement, le caractère de saut de ligne "\n" doit être inséré immédiatement après le dernier caractère de "données<sub>n</sub>".

Les structures prévues pour chaque requête doivent être appliquées d'une manière stricte. Elles assurent une analyse adéquate des messages reçus par les interfaces réseaux du serveur et des clients. L'annexe III présente les structures des messages pour chacune des requêtes et réponses spécifiques. L'annexe IV présente, en guise d'exemples concrets, un ensemble de messages qui correspondent à la plupart des requêtes mentionnées.

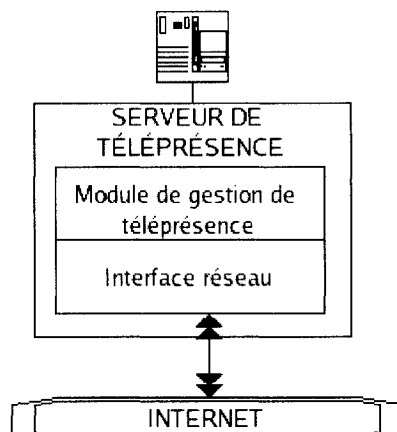
## **CHAPITRE 5 : SERVEUR DE TÉLÉPRÉSENCE MULTI-USAGERS**

La finalité de ce chapitre est de décrire avec plus de précision les composants du serveur de téléprésence multi-usagers. Le serveur est conçu de façon à propager les positions des avatars (têtes et souris 3D des utilisateurs à tous les autres. Un utilisateur spécifique peut ainsi observer les déplacements et les intentions de ses collaborateurs. Les sections suivantes présentent les détails d'implémentation, les principaux composants, le diagramme de classe et la gestion réseautique du serveur.

### **5.1 Description globale du serveur développé**

#### **5.1.1 Détails d'implémentation et principaux composants**

Le serveur de téléprésence est conçu en C++ et il utilise la librairie C++ Sockets Library afin de gérer les communications réseaux. Malgré le fait que la version développée dans le cadre de cette étude fonctionne sur Linux, le serveur est multi-plateforme dans le sens où sa migration sur un autre système d'exploitation ne nécessite qu'une recompilation. Aucune modification du code source n'est nécessaire puisqu'il utilise des librairies C++ standards et que la librairie C++ Sockets Library est multi-plateforme. Afin de faciliter sa compréhension, le serveur peut être séparé en deux composantes principales : le module de gestion de téléprésence et l'interface réseau (figure 5.1).



**Figure 5.1** : Composantes du serveur de téléprésence multi-usagers.

Le rôle du module gestion de téléprésence est simple : il s'occupe de conserver les données d'identification des clients (adresses IP, numéros d'identification, ...) qui participent à une session d'entraînement collaboratif et il s'assure d'une multidiffusion continue des matrices de position et d'orientation des têtes et des souris 3D des usagers à tous les autres (lorsqu'ils sont en mouvement).

Les tâches de nature purement réseautique sont effectuées par l'interface réseau. Elle a pour objectif de recevoir et de gérer les envois UDP des clients qui participent à la session. Le protocole UDP est utilisé afin d'assurer une transmission rapide des messages. La haute fréquence et la nature temporelle des messages envoyés rendent négligeable le fait que l'acheminement n'est pas assuré.

Le module de gestion de téléprésence s'occupe de transmettre, à tous les autres clients identifiés, les données qui décrivent les déplacements des utilisateurs. La transmission de ces données permet à l'utilisateur d'un client d'observer les déplacements des autres utilisateurs, à l'intérieur de la salle de chirurgie virtuelle, via des avatars qui représentent leurs corps et leurs souris 3D. Cette fonctionnalité, qui est complémentaire au système de communication vocale utilisé par les clients, permet une meilleure compréhension des intentions et des actions des participants.

### **5.1.2 Diagramme de classe du serveur**

Le diagramme de classe du serveur de téléprésence est présenté dans l'annexe V. Le serveur est séparé en deux ensembles de classes contenant respectivement les classes de gestion de l'interface réseau et du module de gestion de téléprésence. Les classes du premier ensemble proviennent de la librairie C++ Sockets Library, implémentent le protocole UDP et peuvent être associées à la boîte « Interface réseau » de la figure 5.1. Le deuxième ensemble n'est constitué que d'une seule classe. Celle-ci a été créée afin de jouer le rôle du module de gestion de téléprésence présenté dans la figure 5.1. Les détails et les explications concernant le diagramme de classe sont présentés dans l'annexe V.

## **5.2 Gestion réseautique du serveur**

### **5.2.1 Initiation de la communication**

Les clients sont responsables d'initier la communication avec le serveur. La réception du premier message d'un client génère un numéro d'identification qui est associé à son adresse IP. Le serveur obtient donc, suite à la réception de messages de clients distincts, une table d'identification des utilisateurs inscrits à la session de téléprésence.

### **5.2.2 Processus de transmission des données de téléprésence**

Le déplacement de la tête ou de la souris 3D d'un client implique la transmission au serveur de ses données positionnelles. Un client immobile n'effectue aucune transmission. Le flux de données d'un client spécifique est propagé à tous les autres clients connus du serveur. Les messages transmis incluent le numéro d'identification du client en déplacement. La réception d'un message mentionnant un numéro d'identification inconnu indique au client (récepteur) qu'il s'agit d'un nouveau collaborateur. Un avatar est généré afin de le représenter dans la salle de chirurgie. Tous les messages subséquents reçus, identifiés par le même numéro, sont associés à ce collaborateur et sont utilisés pour effectuer le déplacement de l'avatar concerné.



### 5.2.3 Format et structure des données échangées

La plupart des messages réseaux du serveur de téléprésence utilisent la même structure globale qui a été présenté dans la section 4.4.3 :

ÉTIQUETTE Type\_de\_données<sub>0</sub>:données<sub>0</sub> Type\_de\_données<sub>1</sub>:données<sub>1</sub> ...  
Type\_de\_données<sub>n</sub>:données<sub>n</sub>

où "ÉTIQUETTE" indique la nature de la requête ou de la réponse, "Type\_de\_données<sub>i</sub>:" indique le i<sup>e</sup> type des données qui suivront et "données<sub>i</sub>" représente les i<sup>e</sup> données du type déclaré. Un seul caractère d'espace doit exister entre "ÉTIQUETTE" et "Type\_de\_données<sub>0</sub>". Aucun espace ne doit séparer "Type\_de\_données<sub>i</sub>:" et "données<sub>i</sub>". Un seul caractère d'espace doit séparer les couples "Type\_de\_données<sub>i</sub>:données<sub>i</sub>" et "Type\_de\_données<sub>i+1</sub>:données<sub>i+1</sub>". Finalement, le caractère de saut de ligne "\n" doit être inséré immédiatement après le dernier caractère de "données<sub>n</sub>".

L'annexe VI présente les structures des messages pour chacune des requêtes et réponses spécifiques.

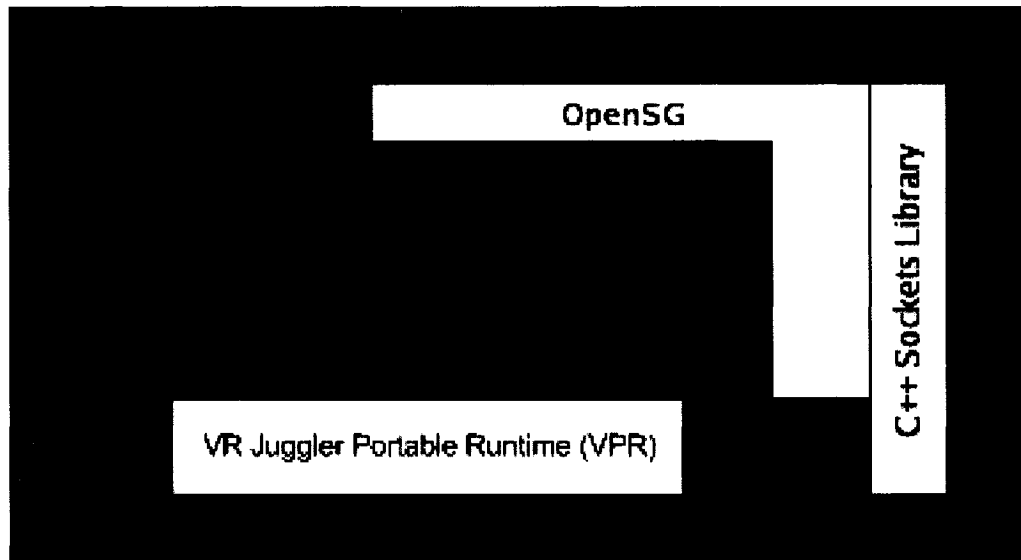
## **CHAPITRE 6 : CLIENT DE SIMULATION DE CHIRURGIE SCOLIOTIQUE EN RÉALITÉ VIRTUELLE**

L'objectif de ce chapitre est de décrire d'une manière plus approfondie le client de simulation de chirurgie scoliotique en réalité virtuelle. Les sections qui suivent présentent le client dans son ensemble et procurent par la suite une description plus détaillée de la gestion de l'interaction de l'utilisateur avec les objets du monde virtuel, de la gestion du graphe de scène, de l'interface avec le rachis scoliotique, de l'interface de téléprésence multi-usagers et du logiciel de communication vocale utilisé.

### **6.1 Description globale du client développé**

#### **6.1.1 Détails d'implémentation et principaux composants**

Le client de simulation de chirurgie scoliotique en réalité virtuelle est conçu en C++ avec la plateforme de développement en réalité virtuelle VR Juggler, le gestionnaire de graphe de scène OpenSG et la librairie réseautique C++ Sockets Library. Le logiciel Skype est utilisé en parallèle afin de permettre une communication vocale entre les utilisateurs. La figure 6.1 montre les différentes couches logicielles sur lesquelles est basé le simulateur.



**Figure 6.1 :** Couches logicielles du client de simulation de chirurgie scoliotique en réalité virtuelle.

La figure montre que VR Juggler est composé de plusieurs entités : CORBA, VPR, Tweek, JCCL et Gadgeteer<sup>17</sup>. La région représentant OpenSG témoigne du fait qu'il est conçu de façon à s'intégrer facilement avec VR Juggler. L'API OpenGL est à la base de VR Juggler et d'OpenSG.

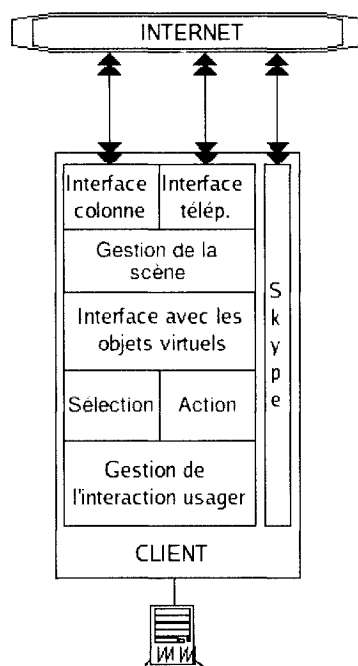
Le prototype actuel est conçu pour le système d'exploitation Linux. Le fait qu'il soit basé sur un ensemble d'outils multi-plateformes implique qu'il est, malgré tout, facile de l'adapter pour d'autres systèmes d'exploitation.

Le client peut être séparé en huit composantes : le module de gestion de l'interaction usager, le module de sélection, le module d'action, l'interface avec les objets virtuels (modèles géométriques), le module de gestion de la scène, l'interface avec le serveur biomécanique de la colonne vertébrale, l'interface avec le serveur de téléprésence et Skype. Il est important de remarquer que cette séparation n'est utilisée que pour faciliter la compréhension du fonctionnement du client. Elle cerne les principales tâches

---

<sup>17</sup> Plus d'information à propos de ces composants est disponible sur le site web de VR Juggler ([www.vrjuggler.org](http://www.vrjuggler.org)).

accomplies, mais son architecture est en fait beaucoup plus complexe. Les réelles entités du client ne peuvent être réduites à une telle simplicité quant à leurs rôles et interactions mutuelles. Le diagramme de classe présenté dans l'annexe VII témoigne d'ailleurs de cette réalité.



**Figure 6.2 :** Composantes du client de simulation de chirurgie scoliotique en réalité virtuelle.

La gestion de l'interaction a pour objectif de traiter les déplacements et les actions de l'utilisateur qui sont exprimés via sa tête (position des lunettes) et la souris 3D (position, états des boutons). Le déplacement de l'utilisateur (dans la CAVE ou avec le clavier) implique un déplacement de sa tête et, bien entendu, des lunettes. Le simulateur utilise la position et l'orientation de celles-ci afin de générer en temps réel une vue sur la scène adaptée à sa perspective.

Le module de sélection détermine, via l'interface avec les objets virtuels, si la souris 3D est dans un état qui permet la sélection d'un objet de la scène. Le module d'action détermine si la souris est dans un état permettant l'exécution d'une action discrète (ex.: appuyer sur le bouton d'une manette) ou continue (ex. : déplacer un implant) sur un ou

plusieurs objets de la scène. Il gère également la mise à jour de la scène lors du déroulement et/ou de la complétion d'une action.

Le module de gestion de la scène s'occupe de l'état du monde virtuel et de son affichage. Il offre un accès aux objets géométriques (ex.: lampe chirurgicale, implants, table d'opération, patient, colonne vertébrale, ...) qui en font partie et permet la création, la modification ou la suppression d'objets. Il est donc possible de modifier la position, l'orientation, la taille et l'apparence des objets de la scène.

L'interface à la colonne gère la connexion et les communications réseautiques avec le serveur biomécanique d'instrumentation du rachis scoliotique. Toute action sur la colonne entraîne l'envoi d'une requête au serveur. La réponse du serveur est ensuite reçue et traitée par ce même module afin de mettre à jour l'état de la colonne et son aspect géométrique.

L'interface de téléprésence gère les communications réseautiques avec le serveur de téléprésence multi-usagers. Lorsqu'un message reçu concerne un participant non-identifié, un avatar est créé et ajouté à la scène. La matrice de transformation de cet avatar est par la suite continuellement mise à jour lors de la réception de messages le concernant. Il est donc possible d'observer le déplacement des collaborateurs.

Le logiciel Skype est utilisé pour la communication inter-collaborateurs. Il est parallèlement installé sur la même machine que le simulateur, mais n'entretient aucun lien avec celui-ci.

### **6.1.2 Diagramme de classe du client**

Le diagramme de classe simplifié du client de simulation est présenté dans l'annexe VII. Le client est séparé en deux ensembles contenant respectivement la majorité des classes de gestion de l'interface réseau et de la simulation graphique de la chirurgie scoliotique. Les classes du premier ensemble proviennent de la librairie C++ Sockets Library et jouent le rôle de l'interface à la colonne et de l'interface de téléprésence de la figure 6.2.

Les classes du deuxième ensemble s'occupent quant à elle de l'affichage de la scène, de la gestion de l'interaction usager à l'intérieur de cette scène et de la mise à jours de la scène. Ces classes jouent donc le rôle des boîtes suivantes de la figure 6.2 : « Gestion de la scène », « Interface avec les objets virtuels », « Sélection », « Action » et « Gestion de l'interaction usager ». Les détails et les explications concernant le diagramme de classe sont présentés dans l'annexe VII.

### **6.1.3 Description générale de la salle de chirurgie virtuelle**

La figure 3.3 du troisième chapitre donne un bon aperçu de la salle de chirurgie virtuelle. La figure 6.3 procure des vues supplémentaires de la scène (prises en mode non-immersif).

Les objets suivants font partie de la salle de chirurgie virtuelle : les murs, le plancher, le plafond, les armoires, le négatoscope, la lampe, la table d'opération, le drap chirurgical, le patient et sa colonne vertébrale, le plateau, les implants, la tige, les outils, les manettes de contrôle de sélection d'implant, les avatars et la manette de contrôle des vues alternatives. La figure 6.4 présente la plupart des modèles VRML utilisés. Les implants chirurgicaux et le rachis scoliotique du patient ne font pas partie de cette figure puisqu'ils ont respectivement été présentés dans les figures 3.6 et 4.3.

Certains des objets de la scène réutilisent des modèles afin de former un tout. La lampe chirurgicale et la manette de sélection d'implant en sont des exemples. La lampe réutilise les modèles 11 et 12 afin de former l'objet final qui est représenté dans la cinquième image de la figure 6.3. La manette de sélection réutilise quant à elle le modèle 22 afin de former l'une des manettes représentées dans la sixième image (ou dans la troisième image de la figure 3.3).

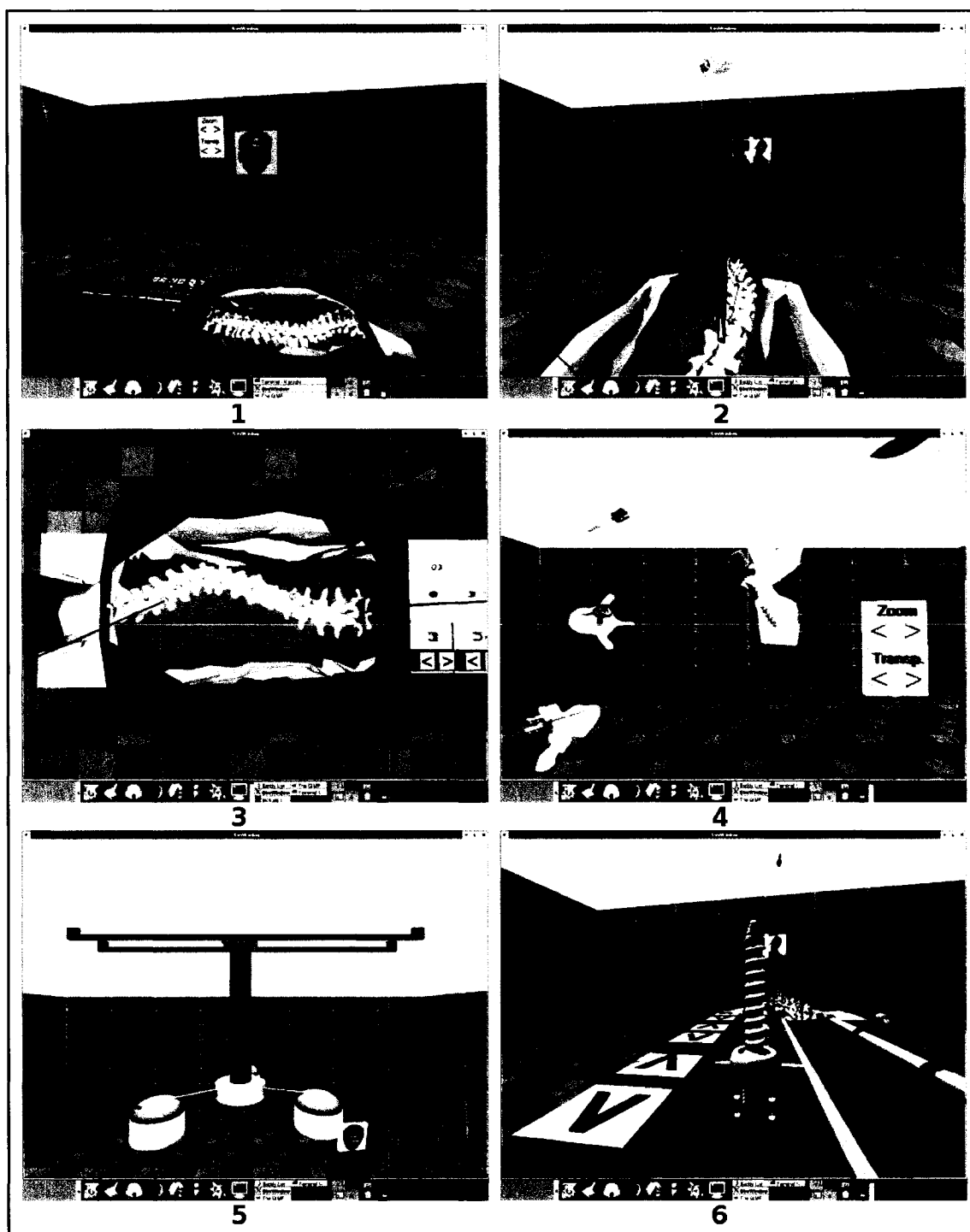


Figure 6.3 : Vues supplémentaires de la salle de chirurgie virtuelle (mode non-immersif).

La scène est globalement éclairée par une lumière de position (« point light ») située près du centre du plafond. Trois lumières de projection (« spot light ») sont positionnées dans les projecteurs de la lampe chirurgicale afin d'obtenir une vue claire et précise de la zone opératoire. La lumière émise par chaque projecteur est réduite à un cône dirigé vers le bas. Elle est à sa plus forte intensité au centre du cône et s'atténue graduellement vers la périphérie. Chaque bras de la lampe peut être déplacé en rotation afin d'offrir un éclairage paramétrable sur la zone d'opération.

Le nombre d'implants (12 crochets, 6 vis pédiculaires fixes et 5 vis pédiculaires multiaxiales) et leur complexité polygonale (entre 20 000 et 30 000 polygones chacun) ont justifié l'utilisation de manettes de sélection d'implant. L'objectif est de réduire à trois le nombre d'implants présents sur le plateau afin d'améliorer les performances d'affichage. Des outils de réduction polygonale ont été utilisés afin de diminuer la précision et le nombre de polygones des modèles, mais les résultats ont été peu concluants : la qualité des modèles se dégradait trop rapidement. L'utilisation de manettes de sélection est donc un compromis permettant à l'utilisateur, via les boutons « précédent » et « suivant », de cycler au travers les trois catégories d'implants avant d'effectuer son choix.

Le modèle 16 de la figure 6.4 correspond à la manette de contrôle qui procure des vues alternatives d'un implant ou d'une vertèbre implantée. Elle permet à l'utilisateur d'observer avec plus de précision un implant ou de revoir la position d'insertion de chaque vis ou crochets au sein des vertèbres. Les boutons augmentent ou diminuent le facteur de grossissement des vues ou rendent une vertèbre plus ou moins transparente. La transparence permet une meilleure évaluation de l'insertion. Quatre vues sont proposées : trois projections orthogonales selon les axes principaux et une vue principale grossie qui possède la même orientation que l'objet reproduit.

Une tige d'instrumentation est initialement déposée sur le plateau à implants/outils. Elle ne fait pas partie de la figure 6.4 étant donné qu'elle est générée à l'interne. Elle est

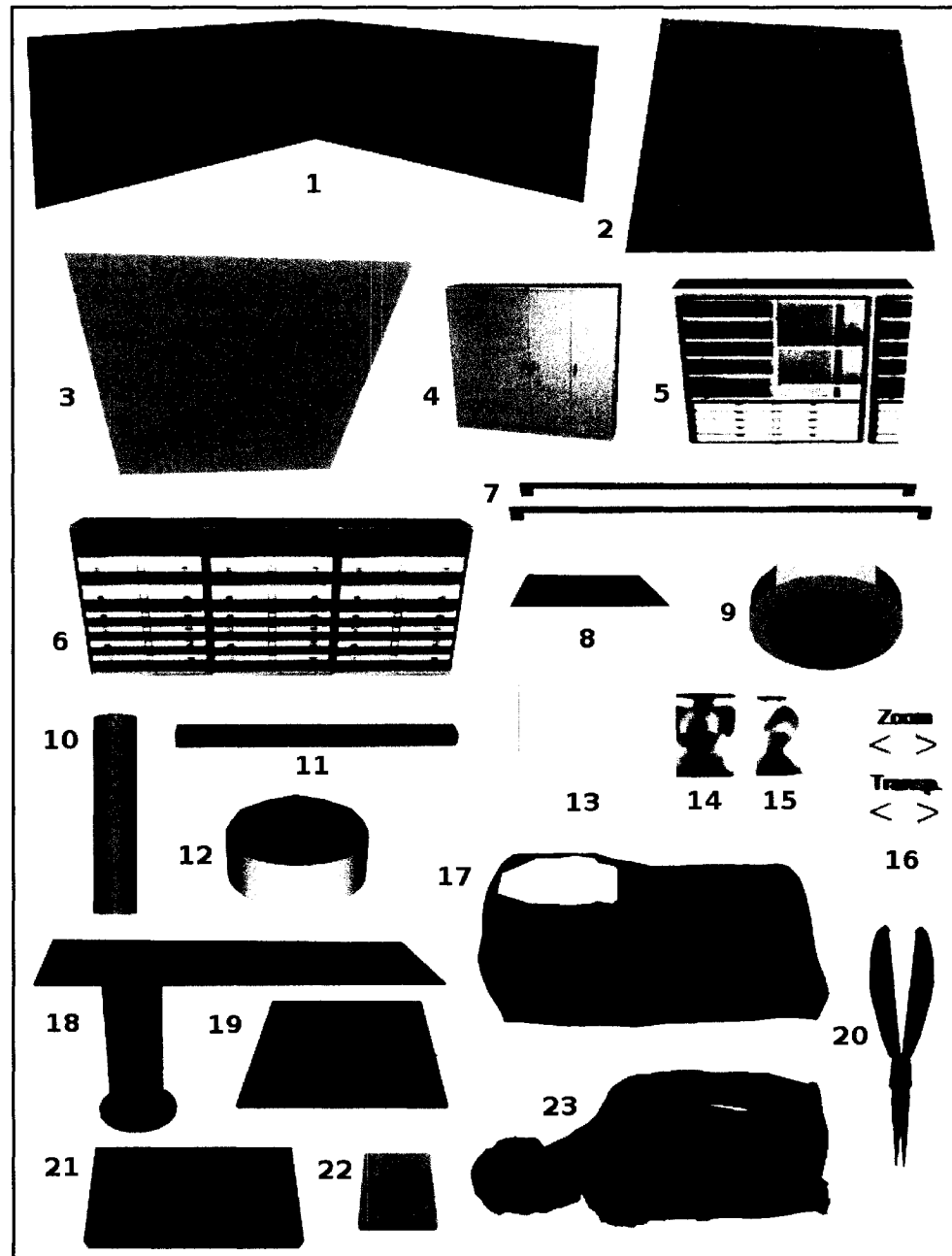


constituée de six segments cylindriques et de sept sphères qui représentent les points de contrôle. Ces points peuvent être utilisés pour déformer et courber la tige.

L'image 20 correspond à l'outil de rotation de tige utilisé. Il est important de remarquer que l'outil réel n'a pas la même apparence. Il s'agit malgré tout d'une représentation adéquate au contexte de cette recherche.

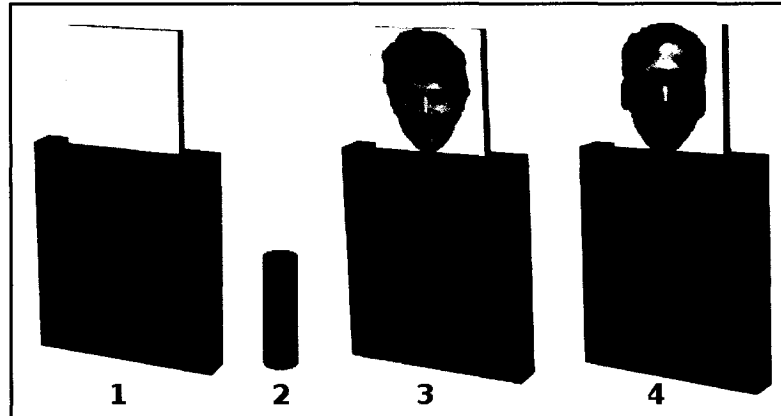
Un écrou est aussi disponible sur le plateau. Bien que non supporté pour l'instant, il pourrait éventuellement être utilisé pour verrouiller un implant à une tige préalablement attachée.

L'image 23 représente le modèle générique du patient utilisé pour toutes les simulations. Le rachis scoliotique reconstruit est normalement aligné et positionné dans l'ouverture dorsale. Le bas du corps du patient a été supprimé afin de diminuer la complexité polygonale. Le fait qu'il soit, de toute façon, caché par le drap chirurgical (image 17) justifie également ce choix.



**Figure 6.4 :** Modèles géométriques de la salle de chirurgie virtuelle : 1- Murs ; 2- Plancher ; 3- Plafond ; 4- Petite armoire ; 5- Première partie de l'armoire en « L » ; 6- Deuxième partie de l'armoire en « L » ; 7- Rails de la lampe ; 8- Train du rail de la lampe ; 9- Base de la lampe qui permet la rotation des bras ; 10- Cylindre qui relie le train et la base de la lampe ; 11- Bras de projecteur (« spot light ») de la lampe ; 12- Projecteur fixé au bras ; 13- Négatoscope ; 14- Radiographie frontale du rachis du patient (fixée sur négatoscope) ; 15- Radiographie latérale du rachis du patient (fixée sur négatoscope) ; 16- Manette de contrôle des vues alternatives ; 17- Drap chirurgical ; 18- Table d'opération ; 19- Plateau à implants/outils ; 20- Outil de rotation de tige ; 21- Corps de la manette de sélection d'implant ; 22- Bouton de la manette de sélection d'implant ; 23- Haut du corps du patient.

Les deux premières images de la figure 6.5 montrent les modèles utilisés pour représenter le corps et la souris 3D d'un participant (son avatar). Des couleurs différentes et des textures peuvent être spécifiées afin de faciliter la reconnaissance inter-collaborateurs.

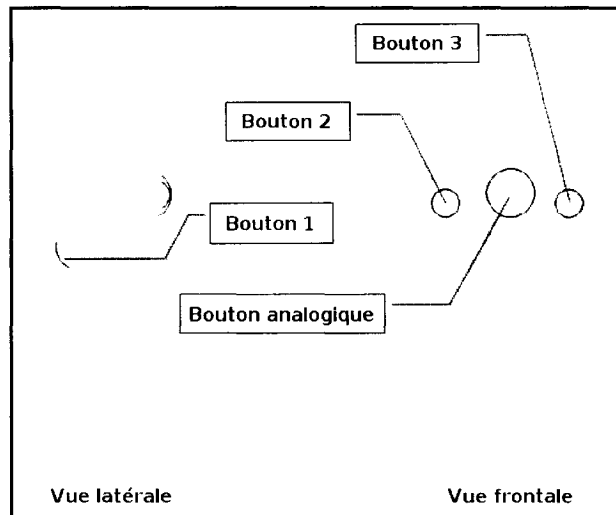


**Figure 6.5 :** Avatars des participants : 1- Le corps et la tête ; 2- La souris 3D ; 3- Avatar bleu avec texture faciale ; 4- Avatar noir avec texture faciale.

## **6.2 Gestion de l'interaction usager**

### **6.2.1 Interaction en mode immersif**

La souris 3D est utilisée pour interagir avec la simulation en mode immersif. La figure 6.6 présente un schéma de la souris et de ses boutons. Bien que non montré dans le schéma, elle comporte également un capteur qui permet d'obtenir sa position et son orientation selon six degrés de liberté.



**Figure 6.6** : Schéma de la souris 3D et de ses boutons.

La souris 3D est considérée comme une extension de la main avec laquelle l'utilisateur peut sélectionner/déplacer des objets et effectuer des actions.

CVrS3 a été conçu de façon à rendre l'interaction la plus simple possible : seuls les deux premiers boutons sont nécessaires. Le premier bouton (qui a l'apparence d'une gâchette) est généralement utilisé pour déplacer un objet sélectionné ou appuyer sur le bouton d'une des manettes. Le deuxième bouton est utilisé pour l'application d'une force de nature chirurgicale. Plus d'informations à ce sujet seront présentées dans la section 6.3.

### 6.2.2 Interaction en mode non-immersif

Le client en mode non-immersif propose une interface avec clavier et souris. Cette interface permet notamment de contrôler la position et l'orientation de la tête de l'utilisateur (le point de vue sur la scène). Elle simule le déplacement des lunettes portées par l'utilisateur en mode immersif. La souris 3D est aussi simulée via cette interface. Un cylindre vert dont la position et l'orientation peuvent être contrôlées la représente dans la scène. Les boutons de la souris standard sont utilisés pour simuler la pression de ses boutons. Les commandes de la tête et de la souris 3D sont présentées dans les tableaux 6.1 et 6.2.

**Tableau 6.1** : Commandes de contrôle de la tête en mode non-immersif.

<b>Action de la tête</b>	<b>Touche(s) du clavier</b>
Déplacement vers l'arrière	2 sur pavé numérique
Déplacement vers la gauche	4 sur pavé numérique
Déplacement vers la droite	6 sur pavé numérique
Déplacement vers l'avant	8 sur pavé numérique
Déplacement vers le bas	7 sur pavé numérique
Déplacement vers le haut	9 sur pavé numérique
Rotation vers le haut	CTRL + 2 sur pavé numérique
Rotation vers la gauche	CTRL + 4 sur pavé numérique
Rotation vers la droite	CTRL + 6 sur pavé numérique
Rotation vers le bas	CTRL + 8 sur pavé numérique
Rotation à sens horaire	1 sur pavé numérique
Rotation à sens anti-horaire	3 sur pavé numérique

**Tableau 6.2** : Commandes de contrôle de la souris 3D en mode non-immersif

<b>Action de la souris 3d</b>	<b>Touche(s) du clavier / Entrée Souris</b>
Déplacement vers l'arrière	ALT + souris vers l'arrière
Déplacement vers l'avant	ALT + souris vers l'avant
Déplacement vers la gauche	CTRL + souris vers la gauche
Déplacement vers la droite	CTRL + souris vers la droite
Déplacement vers le haut	CTRL + souris vers l'avant
Déplacement vers le bas	CTRL + souris vers l'arrière
Rotation vers la gauche	SHIFT + souris vers la gauche

Rotation vers le droite	SHIFT + souris vers la droite
Rotation vers le haut	SHIFT + souris vers l'arrière
Rotation vers le bas	SHIFT + souris vers l'avant
Rotation à sens horaire	Flèche droite
Rotation à sens anti-horaire	Flèche gauche
Bouton 1	Bouton gauche (souris)
Bouton 2	Bouton du milieu (souris)

### 6.3 Interface avec les objets virtuels

Avant d'entamer la présentation des méthodes d'interaction proposées pour CVrS3, il ne faut pas oublier qu'elles sont le résultat d'une intégration du module S3. Les manœuvres proposées sont donc fortement dépendantes des fonctions de S3 qui sont conçues pour une interface conventionnelle. Le résultat est qu'une moins grande liberté est offerte quant aux actions pouvant être accomplies lors d'une vraie chirurgie. Les méthodes d'interaction proposées sont donc relativement simples en comparaison aux possibilités offertes dans le vrai monde.

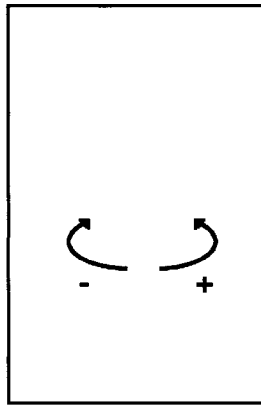
#### 6.3.1 Sélection d'un objet

La sélection d'un objet via la souris 3D est préalable à l'application de toute action. Une sélection s'effectue lorsqu'elle est à une distance suffisante d'un objet sélectionnable. Lorsque plusieurs objets sont à proximité, l'objet le plus près est sélectionné. Une ligne bleue se trace entre la souris et celui-ci. Les objets suivants peuvent être sélectionnés : les trois projecteurs de la lampe chirurgicale, les boutons des manettes de sélection d'implant, les implants, les points de contrôle des tiges, l'outil de dérotation, les boutons de la manette de vues alternatives et les implants insérés dans les vertèbres.

### 6.3.2 Positionnement des projecteurs de la lampe chirurgicale

Chaque projecteur de la lampe peut être déplacé en rotation (360 degrés) afin de modifier l'éclairage de la zone opératoire. La procédure à appliquer est simple :

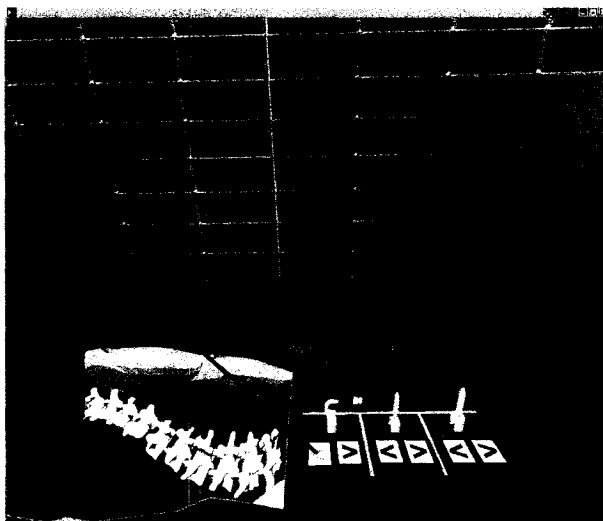
1. Sélectionner un projecteur (voir l'image 5 de la figure 6.3) ;
2. Maintenir le bouton 1 enfoncé et pivoter la souris 3D latéralement dans le sens horaire ou anti-horaire (figure 6.7).



**Figure 6.7** : Rotation de la souris 3D à appliquer pour déplacer un projecteur de la lampe.

### 6.3.3 Choix d'un implant à l'aide d'une manette de sélection

Pour voir le prochain implant ou le précédent d'une catégorie, il faut d'abord sélectionner le bouton à droite ou à gauche de la manette concernée. Chaque clic subséquent du bouton 1 provoque l'affichage de l'implant suivant ou précédent (figure 6.8).



**Figure 6.8 :** Clic du bouton « précédent » de la manette de sélection de crochets.

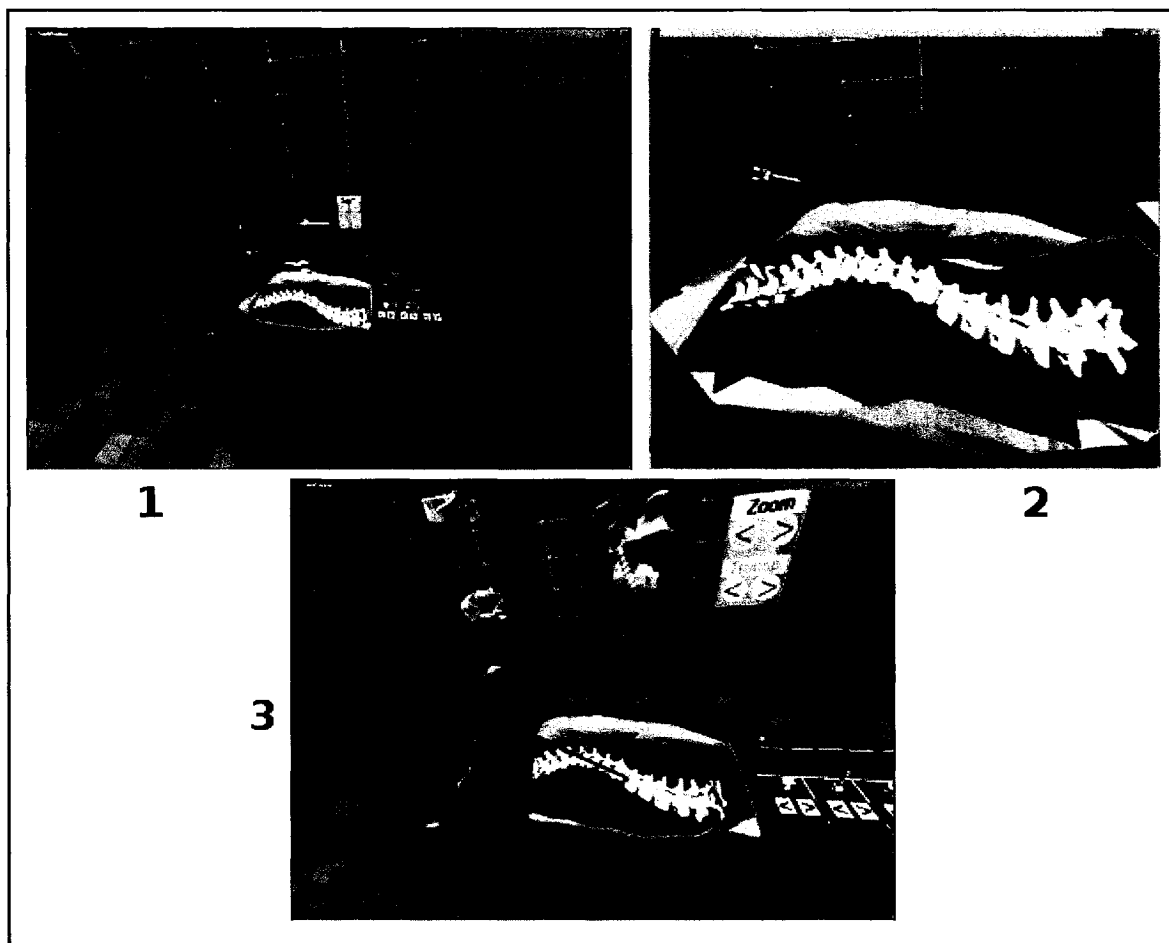
#### **6.3.4 Insertion d'un implant**

Pour insérer un implant dans une vertèbre, il faut :

1. Sélectionner un implant avec la souris 3D ;
2. Déplacer l'implant (« drag »), en maintenant le bouton 1 enfoncé, jusqu'à ce qu'il soit assez près du côté de la vertèbre à implanter (figure 6.9).

Le maintien d'une pression positionne l'implant devant la souris. La ligne de sélection demeure visible. Les déplacements et les rotations de la souris sont appliqués sur l'implant de façon à ce qu'il demeure face à celle-ci. Lorsque l'implant est assez près d'un des côtés d'une vertèbre (gauche ou droite), il s'insère automatiquement. Les crochets sont automatiquement insérés à la position supra laminaire pré-calculée pour le côté de la vertèbre concernée. Les vis pédiculaires sont insérées en position standard (voir tableau 4.1).





**Figure 6.9 :** Insertion d'un implant.

### **6.3.5 Modification de la position d'insertion d'un implant**

Pour modifier la position d'insertion d'un implant, il faut :

1. Sélectionner l'implant préalablement inséré ;
2. Cliquer le bouton 1.

Chaque clic positionne l'implant au point d'insertion suivant parmi l'ensemble des possibilités. Les points d'insertion possibles dépendent du type d'implant et du numéro de la vertèbre. Les règles d'insertion ont été décrites dans le tableau 4.1.

### **6.3.6 Suppression d'un implant**

Pour supprimer un implant, il faut :

1. Sélectionner l'implant préalablement inséré ;
2. Maintenir une pression sur le bouton 1 pendant au moins deux secondes.

Lorsqu'une pression est maintenue, l'implant retourne dans l'état où il peut être déplacé (avec la souris 3D). L'utilisateur peut ainsi effectuer une nouvelle insertion rapidement. Le relâchement du bouton détruit l'implant.

### **6.3.7 Modifier les paramètres des vues alternatives**

La manette des vues alternatives affiche toujours le dernier implant sélectionné. Lorsqu'il s'agit d'un implant inséré, la vertèbre est aussi affichée. La manette est composée de quatre boutons. Le bouton en haut à gauche diminue le facteur de grossissement des vues tandis que celui à droite l'augmente. Le bouton en bas à gauche diminue le facteur de transparence tandis que celui à droite l'augmente. Pour modifier un des paramètres, il faut sélectionner le bouton approprié et cliquer sur le bouton 1 de la souris 3D jusqu'à ce que la vue désirée soit affichée.

### 6.3.8 Modifier la courbure d'une tige

La tige d'instrumentation est composée de sept points de contrôle. La sélection d'un point et le maintien d'une pression sur le bouton 1 permettent de déplacer la tige. L'utilisation du bouton 2 (pour appliquer une force) déplace uniquement le point et modifie la position et l'orientation du ou des segments y étant connectés. L'utilisateur peut ainsi altérer la courbure globale de la tige en répétant cette action sur les points de contrôle appropriés (figure 6.10).

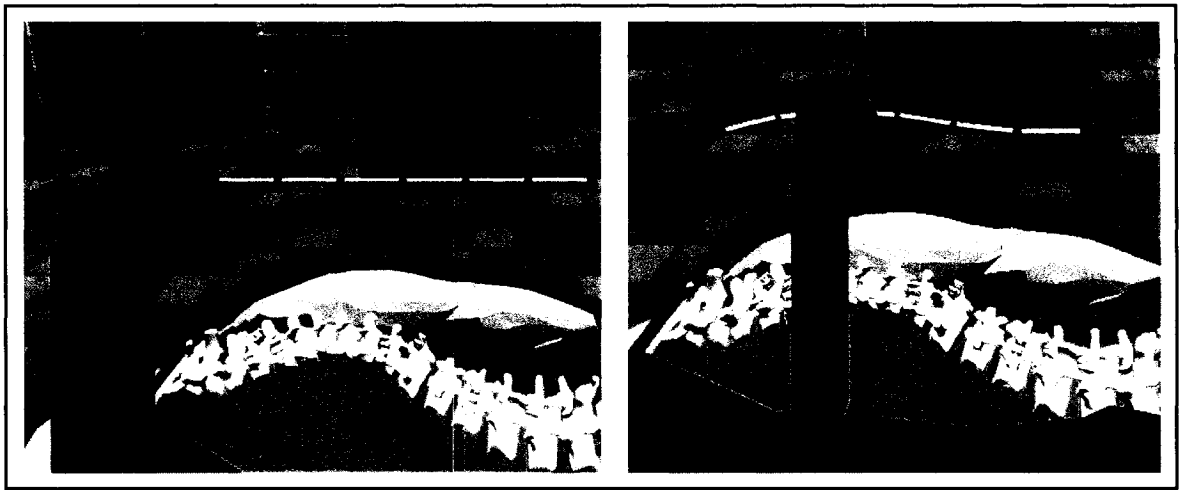


Figure 6.10 : Modification de la courbure d'une tige.

### 6.3.9 Méthodes d'interaction proposées pour les manœuvres de S3

Cette section présente les méthodes d'interaction proposées pour adapter les manœuvres de correction chirurgicale de S3 dans l'environnement virtuel de CVrS3.

#### Attachement et détachement d'une tige

Avec S3, une tige peut être attachée d'un seul coup en spécifiant l'ordre séquentiel des implants à attacher. Il est aussi possible d'attacher un seul ou quelques implants à la fois en spécifiant l'ordre d'attachement. CVrS3 évite l'utilisation de menus afin de créer une expérience se rapprochant le plus possible d'une intervention réelle. Le fait d'éviter volontairement ce type d'interface complexifie les méthodes à utiliser. Ce désavantage n'est par contre pas trop dramatique puisque les chirurgiens attachent généralement les

implants d'un seul coup (avec S3). Ils utilisent aussi, la plupart du temps, un ordre d'attachement conventionnel (de haut en bas ou de bas en haut). Les méthodes d'attachement suivantes sont donc proposées :

**Tableau 6.3** : Méthodes d'attachement proposées pour CVrS3.

<b>Attachement d'un seul coup</b>	<b>Du côté gauche</b>	De haut en bas	L'utilisateur <b>déplace la tige</b> à partir d'un des <b>points de contrôle du haut</b> et l'approche du <b>côté gauche de la colonne</b> .
		De bas en haut	L'utilisateur <b>déplace la tige</b> à partir d'un des <b>points de contrôle du bas</b> et l'approche du <b>côté gauche de la colonne</b> .
	<b>Du côté droit</b>	De haut en bas	L'utilisateur <b>déplace la tige</b> à partir d'un des <b>points de contrôle du haut</b> et l'approche du <b>côté droit de la colonne</b> .
		De bas en haut	L'utilisateur <b>déplace la tige</b> à partir d'un des <b>points de contrôle du bas</b> et l'approche du <b>côté droit de la colonne</b> .
<b>Attachement implant par implant</b>	L'utilisateur déplace la tige à proximité de la colonne (sans qu'elle soit assez près pour être attachée). En maintenant une pression sur le bouton 2 de la souris 3D, il crée un lien (une ligne se trace entre la tige et la souris) entre la tige et l'implant auquel il veut l'attacher. Il peut attacher la tige aux autres implants en procédant de la même façon. <b>Remarque:</b> le premier implant auquel la tige est attachée détermine le côté de la colonne qui sera utilisé tout au long du processus d'attachement. Il est impossible d'attacher une tige à des implants se situant de l'autre côté de la colonne.		
<b>Détachement d'un seul coup</b>	L'utilisateur sélectionne la tige et maintient une pression sur le bouton 1 de la souris 3D pendant au moins deux secondes. <b>Remarque:</b> le détachement ne s'effectue que pour les implants non verrouillés.		
<b>Détachement implant par implant</b>	L'utilisateur sélectionne l'implant qu'il veut détacher et maintient une pression sur le bouton 1 de la souris 3D pendant au moins deux secondes. <b>Remarque:</b> un implant verrouillé ne peut être détaché. Effectuer un détachement sur un tel implant revient à le déverrouiller (retirer l'écrou y étant vissé).		

### Rotation d'une tige

La rotation d'une tige est normalement appliquée à partir de son centre. S3 effectue ses calculs en tenant compte de cette propriété. La procédure proposée est la suivante :

1. Sélectionner et déplacer l'outil de rotation à proximité du centre d'une tige attachée. L'outil se positionne automatiquement de façon à permettre une rotation ;
2. Maintenir une pression sur le bouton 2 de la souris 3D tout en la déplaçant vers l'avant (pousser) ou vers l'arrière (tirer). Un déplacement vers l'avant génère une rotation positive tandis qu'un déplacement opposé génère une rotation négative.

La rotation subie par l'outil est calculée en fonction de la distance parcourue par la souris. L'équivalence distance/angle est définie de façon à ce qu'une grande distance soit requise pour générer un petit angle. L'objectif est de créer l'impression qu'une grande force est nécessaire. Le fait que les positions de la souris obtenues par le système de repérage ne soient pas très précises justifie également ce choix. Ainsi, un déplacement trop rapide ou erroné ne provoque pas l'envoi de requêtes inutiles au serveur biomécanique.

Une nouvelle requête de rotation est envoyée à chaque fois que l'angle de rotation subie par l'outil dépasse positivement ou négativement une valeur d'incrément d'angle prédéterminée. L'algorithme suivant décrit la gestion de l'envoi des requêtes avec plus de précision :

```

 $\Theta_{ref} = \Theta_{act}$ 
TANT QUE (Le bouton 2 est pressé)
     $\Delta_{ang} = \Theta_{act} - \Theta_{ref}$ 
    SI ( $\Delta_{ang} > \Theta_{inc}$ )
         $\Theta_{ref} = \Theta_{ref} + \Theta_{inc}$ 
        Envoie d'une requête de rotation de  $\Theta_{inc}$ 
    SINON SI ( $\Delta_{ang} < -\Theta_{inc}$ )
         $\Theta_{ref} = \Theta_{ref} - \Theta_{inc}$ 
        Envoie d'une requête de rotation de  $-\Theta_{inc}$ 
    FIN SI
FIN TANT QUE

```

où  $\Theta_{act}$  est l'angle actuel de l'outil par rapport à la tige,  $\Theta_{ref}$  est l'angle de référence de l'outil par rapport à la tige,  $\Delta_{ang}$  est la différence entre ces angles et  $\Theta_{inc}$  est l'incrément d'angle nécessaire à l'envoi d'une requête au serveur biomécanique.

### **Verrouillage d'une tige**

Le verrouillage d'une tige s'effectue en vissant des écrous sur les implants. Les implants de CD Horizon n'ont que deux états: verrouillé ou déverrouillé. La méthode proposée consiste à sélectionner (à partir du plateau) et déplacer un écrou à proximité de l'implant à verrouiller. Pour déverrouiller un implant, il suffit de le sélectionner et de maintenir une pression sur le bouton 1 de la souris pendant au moins deux secondes.

### **Compression/distraktion de vertèbres**

La compression/distraktion de vertèbres implique la sélection de deux implants (attachés à la même tige) sur deux vertèbres différentes. Le chirurgien utilise normalement ses deux mains pour effectuer une telle manoeuvre. La contrainte d'utilisation d'une seule souris 3D augmente la complexité des méthodes d'interaction pouvant être utilisées. La procédure suivante est proposée :

1. Déplacement du premier outil de compression/distraktion sur le premier implant ;
2. Déplacement du deuxième outil de compression/distraktion sur le deuxième implant. Une flèche (objet géométrique) se trace entre les deux implants ;
3. Sélectionner un des outils et maintenir une pression sur le bouton 2 de la souris 3D tout en la déplaçant vers la gauche/droite pour compresser/distracter les vertèbres.

Un déplacement vers la droite avec l'outil de droite génère une distraktion tandis qu'un déplacement vers la gauche génère une compression. Ce comportement est inversé lorsque l'outil de gauche est utilisé. La longueur de la flèche est calculée en fonction de la distance parcourue par la souris. La relation entre ces distances est régie de façon à ce que le déplacement de la souris génère une translation intervertébrale de moindre amplitude.

Une nouvelle requête de compression/distraktion est envoyée au serveur biomécanique à chaque fois que la longueur de la flèche dépasse positivement ou négativement une valeur d'incrément de translation prédéterminée. L'algorithme suivant décrit la gestion de l'envoi des requêtes avec plus de précision :

```

 $d_{ref} = d_{act}$ 
TANT QUE (Le bouton 2 est pressé)
     $\Delta_{dis} = d_{act} - d_{ref}$ 
    SI ( $\Delta_{dis} > d_{inc}$ )
         $d_{ref} = d_{ref} + d_{inc}$ 
        Envoie d'une requête de rotation de  $d_{inc}$ 
    SINON SI ( $\Delta_{dis} < -d_{inc}$ )
         $d_{ref} = d_{ref} - d_{inc}$ 
        Envoie d'une requête de rotation de  $-d_{inc}$ 
    FIN SI
FIN TANT QUE

```

où  $d_{act}$  est la longueur actuelle de la flèche,  $d_{ref}$  est la distance de référence entre les deux implants,  $\Delta_{dis}$  est la différence entre ces distances et  $d_{inc}$  est l'incrément de translation nécessaire à l'envoi d'une requête au serveur biomécanique.

### Dérotation vertébrale directe

Cette manœuvre de S3 permet de modifier l'angle de rotation d'une vertèbre en choisissant les implants (attachés à la même tige) via lesquels la force doit être appliquée. La procédure suivante est proposée :

1. Déplacer un ou plusieurs outils de dérotation sur le ou les implants à utiliser. Une flèche se trace entre les implants lorsque plusieurs outils sont utilisés ;
2. Sélectionner la vertèbre qui doit subir une dérotation à l'aide de la souris et appuyer sur le bouton 1. La vertèbre devient rouge ;
3. Sélectionner un des outils (si plusieurs sont utilisés) et maintenir une pression sur le bouton 2 tout en poussant/tirant vers l'avant/arrière. Un déplacement vers l'avant génère une rotation positive tandis qu'un déplacement opposé génère une rotation négative.

La rotation subie par le(s) outil(s) est calculée en fonction de la distance parcourue par la souris. L'équivalence distance/angle est définie de façon à qu'une grande distance soit requise pour générer un petit angle.

Une nouvelle requête de dérotation vertébrale directe est envoyée à chaque fois que l'angle de rotation subi par l'outil dépasse positivement ou négativement une valeur d'incrément d'angle prédéterminée. L'algorithme gérant l'envoi des requêtes est fort semblable à celui qui est utilisé pour une rotation de tige.

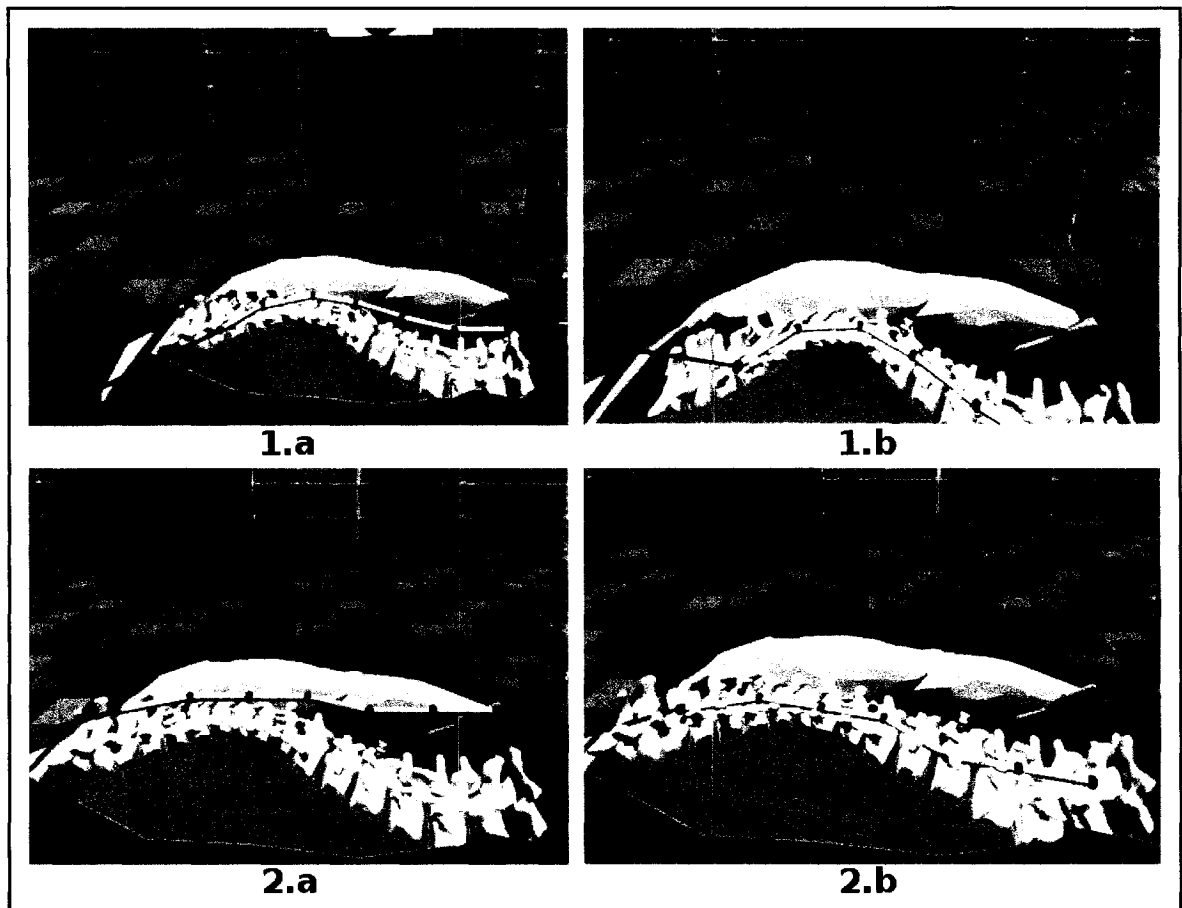
### **6.3.10 Méthodes d'interaction et manœuvres implantées**

La grande envergure du projet et son contexte ont limité le nombre de manœuvres de correction chirurgicale de S3 ayant pu être implantées. Le fait qu'il s'agisse d'une étude dont l'objectif est d'évaluer la faisabilité du développement d'un simulateur en réalité virtuelle collaboratif intégrant le moteur biomécanique de S3, et non d'un travail de développement d'un produit commercial, justifie également cette limitation. Les sous-sections suivantes présentent les méthodes d'interaction à utiliser pour les manœuvres implantées.

#### **Attachement d'une tige**

Le prototype développé permet l'attachement complet d'une tige selon la méthode proposée dans la section 6.3.9. L'utilisateur n'a qu'à déplacer une tige (via un des points de contrôle) préalablement courbée à proximité d'un des implants auquel il veut l'attacher. L'enlignement de la tige relatif au début et à la fin du rachis est conservé. La figure 6.11 présente deux tentatives d'attachement différentes. Le premier attachement est un échec : la courbure du rachis est encore plus prononcée. Le deuxième attachement provoque quant à lui un meilleur résultat.





**Figure 6.11** : Deux tentatives d'attachement de tige.

Il arrive parfois que le serveur biomécanique ne réussisse pas à converger vers une solution. Cette situation est la plupart du temps liée à l'attachement d'une tige n'ayant pas une courbure normale (trop prononcée et/ou irréaliste) d'un point de vue chirurgical. La force à appliquer pour attacher ce type de tige est généralement trop grande et provoquerait, de toute façon, l'arrachement des implants et/ou la mort du patient. Lorsqu'aucune solution n'est trouvée, la tige est déplacée vers le haut afin d'informer l'utilisateur.

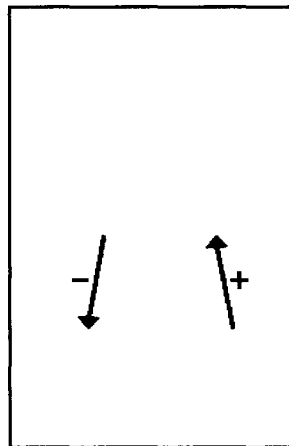
### Détachement d'une tige

Le prototype développé permet le détachement complet d'une tige tel que présenté dans la section 6.3.9. La procédure implantée est par contre légèrement différente : il faut sélectionner un implant fixé sur la tige, plutôt que la tige en soit, avant de maintenir une pression sur le bouton 1 pendant deux secondes.

### Rotation d'une tige

La procédure à utiliser pour la rotation de tige est identique à celle présentée dans la section 6.3.9. La valeur d'incrément d'angle ( $\theta_{inc}$ ) mentionnée dans l'algorithme est fixée à 8 degrés. Des tests qualitatifs ont permis de déterminer qu'il s'agissait du meilleur incrément par rapport au temps de réponse du serveur biomécanique et à la fluidité de la simulation.

Pour appliquer une force, il faut maintenir une pression sur le bouton 2 et déplacer la souris 3D vers l'avant ou vers l'arrière. Un mouvement vers l'avant génère une rotation positive tandis qu'un mouvement opposé génère une rotation négative (figure 6.12).

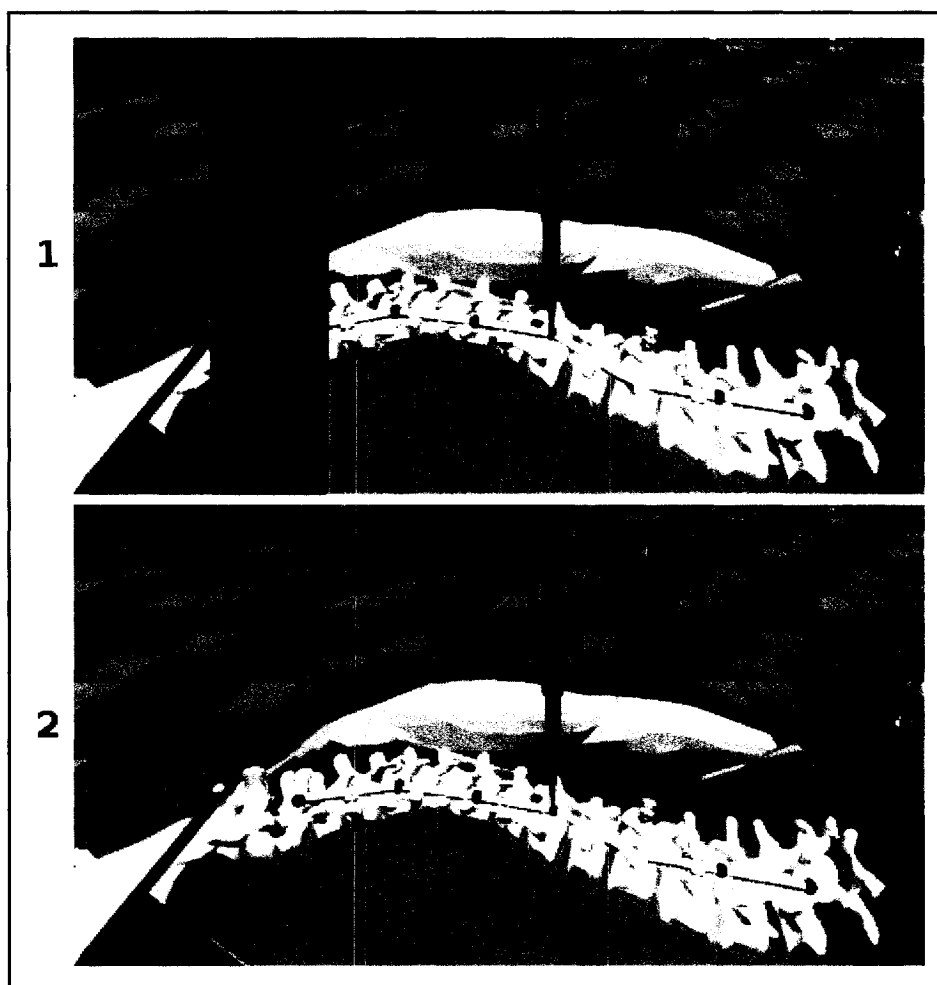


**Figure 6.12** : Déplacement de la souris 3D à appliquer pour une rotation de tige.

L'outil de rotation redevient perpendiculaire au dos du patient lorsque le bouton 2 est relâché. L'outil peut encore être utilisé après ce changement d'état. Ce comportement

simule la situation où le chirurgien repositionne l'outil, par rapport à la tige, afin d'être en meilleure position pour appliquer une force.

La figure 6.13 montre l'état du rachis avant et après l'application d'une faible force vers l'avant. Ces captures d'écran ne montrent malheureusement pas la rotation de l'outil en cours de manœuvre. La première image montre la position de l'outil juste avant l'application d'une force et la deuxième le représente une fois la pression relâchée.



**Figure 6.13 :** Application d'une manœuvre de rotation de tige.

### 6.3.11 Temps de réponse aux actions de l'utilisateur

Les conséquences aux actions suivantes sont immédiatement visibles : déplacement du point de vue de l'utilisateur et de la souris 3D, déplacement d'un projecteur de la lampe, utilisation des manettes, modification de la courbure d'une tige, sélection et déplacement des implants, d'une tige et de l'outil de rotation. Ces actions ont ceci en commun : ils n'ont pas à communiquer avec le serveur biomécanique.

Les tâches d'instrumentation nécessitent l'envoi de requêtes au serveur biomécanique. Le tableau suivant présente une approximation des délais préalables à la visibilité des résultats de chaque tâche. Les tests ont été effectués lors d'une session individuelle en mode non-immersif. Mentionnons que les délais des transmissions réseaux sont négligeable : le client et le serveur biomécanique faisaient partie du même réseau local. Les situations où aucune réponse n'est obtenue sont exclues du tableau.

**Tableau 6.4** : Temps de réponses des tâches d'instrumentation.

Action	Temps de réponse
Insertion d'un implant	~ ½ seconde
Modification de la position d'un implant	~ ½ seconde
Suppression d'un implant	~ ½ seconde
Attachement d'une tige	Entre 1 et 8 secondes
Détachement d'une tige	Immédiat.
Rotation d'une tige (un seul incrément d'angle)	Entre 1 et 5 secondes.

Les délais présentés résultent des traitements effectués par le module S3. Le délai des tâches d'insertion, modification et suppression d'implant est relativement court et prévisible. Le détachement d'une tige est immédiat puisque le client n'attend pas de réponse du serveur avant de modifier la scène.

Les délais pour l'attachement d'une tige varient énormément. En général, plus le nombre d'implants attachés est élevé et plus le temps de réponse sera long. Le type de courbure de la tige (réaliste/irréaliste) joue également un rôle important.

Les délais pour la rotation d'une tige (d'un seul incrément) varient aussi énormément. Ils dépendent généralement de l'état d'instrumentation de la colonne et de la rotation préalable de la tige. Une grande rotation nécessite un traitement séquentiel des requêtes envoyées (une pour chaque incrément). Les résultats intermédiaires sont visibles pour chaque étape, mais le résultat final n'est affiché qu'après la complétion de l'ensemble des requêtes envoyées.

À ces problèmes de délais viennent s'ajouter les situations où le module S3 ne converge pas vers une solution. Le temps nécessaire avant d'arriver à cette constatation est généralement supérieur à 8 secondes. Ces échecs sont souvent liés à la soumission de requêtes irréalistes d'un point de vue chirurgical. L'attachement d'une tige est fortement sujet à ce problème. Le fait que les manœuvres subséquentes sont grandement affectées par l'irréalisme d'une tige rend ce problème encore plus important. Une solution pourrait peut-être diminuer le nombre d'occurrences où cette situation se produit : la création d'un module de validation du réalisme de la forme d'une tige

Les traitements exécutés pour modéliser le comportement du rachis scoliotique nécessitent une très forte puissance de calcul. La faible performance de l'ordinateur utilisé pour héberger le serveur est en grande partie responsable des délais impliqués. L'utilisation d'un ordinateur plus performant pourrait donc réduire de beaucoup le temps de réponse aux requêtes biomécaniques. Une modification du code source du module S3 pourrait aussi améliorer cet aspect. En procurant des fonctions paramétrables sur la précision des résultats, il serait possible d'adapter et de réduire de beaucoup le temps nécessaire à l'obtention de réponses.

## **CHAPITRE 7 : SIMULATION DE L'INSTRUMENTATION DU RACHIS SCOLIOTIQUE**

Ce chapitre décrit les scénarios d'utilisation typique du simulateur en mode individuel et collaboratif. Il présente ensuite le déroulement et les résultats des tests usagers entrepris pour évaluer le prototype développé et déterminer si ce type de simulateur peut ou pourra être utilisé avec efficacité pour l'entraînement des chirurgiens orthopédistes.

### **7.1 Usage individuel du simulateur**

L'utilisateur peut s'entraîner à l'instrumentation du rachis scoliotique en mode immersif ou non-immersif. Un entraînement typique est généralement constitué des étapes suivantes :

1. Le participant observe autour de lui afin de se repérer à l'intérieur de la salle de chirurgie virtuelle et d'identifier les objets en faisant partie ;
2. Le participant déplace un ou plusieurs projecteurs de la lampe chirurgicale afin d'ajuster l'éclairage de la zone opératoire selon ses préférences ;
3. Le participant observe le rachis du patient et les radiographies associées afin d'évaluer la gravité du cas, se faire une représentation mentale du problème orthopédique et prévoir sa stratégie d'instrumentation ;
4. Le participant choisit et insère des implants sur les deux côtés de la colonne en se référant aux vues alternatives pour guider son choix d'implants et évaluer les positions d'insertion. Il peut supprimer et repositionner certains implants lors de ce processus ;
5. Le participant sélectionne, déplace et modifie la courbure d'une tige d'instrumentation à l'aide de ses points de contrôle ;

6. Le participant déplace la tige à proximité des implants situés sur le côté de la colonne qu'il veut attacher. Il observe le résultat de l'attachement et répète l'étape 5 et 6 s'il n'est pas satisfait ou si le serveur biomécanique ne trouve pas de solution ;
7. Le participant utilise l'outil de rotation de tige afin d'appliquer une force tout en observant le résultat.

## **7.2 Usage collaboratif du simulateur**

Cette section décrit deux situations générales où le simulateur peut être utilisé de manière collaborative : l'entraînement collaboratif et le télé-enseignement. Des utilisations intermédiaires peuvent facilement être imaginées. La catégorisation utilisée a comme objectif de donner un aperçu des usages potentiels.

### **7.2.1 Entraînement collaboratif**

Un entraînement collaboratif est généralement effectué par des participants situés à des endroits géographiquement éloignés. L'objectif est de permettre aux participants de collaborer dans la recherche d'une solution optimale à un problème orthopédique spécifique. Il peut s'agir d'une session de planification préopératoire pour un patient donné ou d'une session d'apprentissage général où plusieurs collègues peaufinent mutuellement leurs techniques.

Ce type d'entraînement implique un rôle actif des participants. L'utilisation d'un environnement immersif est donc souhaitable quoique non obligatoire. Le scénario d'utilisation du simulateur est semblable à celui ayant été présenté dans la section précédente. Les principales distinctions sont :

- Les actions accomplies sont réparties parmi l'ensemble des participants ;

- Les participants peuvent communiquer entre eux afin de trouver une solution jugée optimale ;
- Chaque participant peut observer le comportement et les conséquences des actions des autres participants.

### **7.2.2 Télé-enseignement**

Une session de télé-enseignement se distingue d'un entraînement collaboratif dans le sens où elle est constituée de participants dont les rôles sont distincts. Elle implique la participation d'individus ayant un rôle actif ou passif. Un utilisateur actif participe à l'exécution de la chirurgie virtuelle. Un participant passif joue le rôle d'observateur. Un enseignant expérimenté en chirurgie est un exemple de participant actif. Des médecins résidents en cours d'apprentissage peuvent être considérés comme des participants passifs.

Afin d'obtenir un meilleur contrôle, il est préférable qu'un participant actif utilise la version immersive. Un participant passif peut par contre utiliser la version non-immersive sans subir de grandes limitations quant à l'objectif de sa participation : l'apprentissage. Il peut donc assister à une chirurgie virtuelle à partir d'un ordinateur conventionnel (à la maison par exemple), écouter les commentaires du chirurgien en cours de chirurgie et lui poser des questions lorsque nécessaire.

## **7.3 Tests usagers**

Des tests usagers ont été conduits à la fin de l'été 2006 pour évaluer le prototype développé par rapport à son réalisme global, sa facilité d'utilisation, au sentiment d'immersion qu'il procure, au réalisme des tâches chirurgicales pouvant être accomplies et à l'efficacité des fonctionnalités collaboratives implantées. L'objectif est également d'évaluer si ce type de simulateur pourrait être utilisé efficacement en contexte clinique.



### 7.3.1 Description des participants

Un échantillon de cinq participants a été composé. Ceux-ci peuvent être séparés en deux groupes :

1. **Les spécialistes en orthopédie** : groupe formé d'un orthopédiste et d'un résident en orthopédie. Ils ont une forte connaissance de la scoliose idiopathique et de la chirurgie d'instrumentation du rachis scoliotique. Ils ont peu ou pas d'expérience préalable avec S3, l'environnement de la CAVE et les jeux vidéos ;
2. **Les spécialistes en environnements virtuels (EV)** : groupe formé de trois personnes ayant une bonne expérience avec les environnements virtuels, la CAVE et/ou les jeux vidéos. Deux d'entre eux détiennent une maîtrise en génie informatique dont le sujet de recherche était de nature infographique. Ils ont une faible connaissance de la scoliose idiopathique et de sa chirurgie.

Les membres du premier groupe offriront un regard médical sur la pertinence de l'outil développé tandis que les membres du second procureront un regard de nature technique. Il faut remarquer que le nombre de participants est très réduit. L'envergure relativement petite du projet et la difficulté à trouver des spécialistes en orthopédie, qui sont très occupés et peu disponibles, est grandement responsable du faible échantillon. Nous jugeons néanmoins qu'ils contribueront à cerner, modestement, les points forts et faibles du prototype et qu'ils pourront nous donner un bon aperçu de l'utilité que pourrait avoir ce type de simulateur pour la communauté médicale.

### 7.3.2 Mise en place et déroulement des tests

Les tests ont eu lieu au local de la CAVE de l'École Polytechnique de Montréal. Les participants étaient invités à venir en groupe de deux à l'exception de l'un d'eux. Le concepteur du prototype était également présent afin d'assurer le bon déroulement des tests. Le local était aménagé de façon à offrir deux postes de travail relativement isolés

par des paravents : un ordinateur conventionnel et la CAVE. Les deux postes étaient équipés de casques d'écoute avec micro.

Les tests étaient généralement constitués des étapes suivantes :

1. Présentation du projet de recherche, de la scoliose idiopathique, de la chirurgie d'instrumentation du rachis scoliotique et de l'objectif des tests usagers.
2. Présentation du poste de travail utilisé pour le mode non-immersif ;
3. Présentation de l'environnement de la CAVE (utilisée pour le mode immersif) ;
4. Courte démonstration du simulateur à l'intérieur de la CAVE ;
5. Essai individuel du premier participant en mode immersif. L'objectif est de suivre la procédure présentée dans la section 7.1 ;
6. Essai individuel du deuxième participant en mode immersif. L'objectif est de suivre la procédure présentée dans la section 7.1 ;
7. Courte présentation de l'aspect collaboratif du simulateur ;
8. Courte démonstration du simulateur en mode non-immersif ;
9. Premier essai collaboratif des participants. L'un des participants utilise la CAVE tandis que l'autre est sur le poste de travail conventionnel. Ils communiquent entre eux via les casques d'écoute et ont pour objectif de collaborer dans l'accomplissement de la chirurgie. Ils doivent suivre la procédure présentée dans la section 7.1 ;
10. Deuxième essai collaboratif. Les rôles des participants sont inversés ;
11. Recueil des commentaires des participants et distribution des formulaires qu'ils devront remplir.

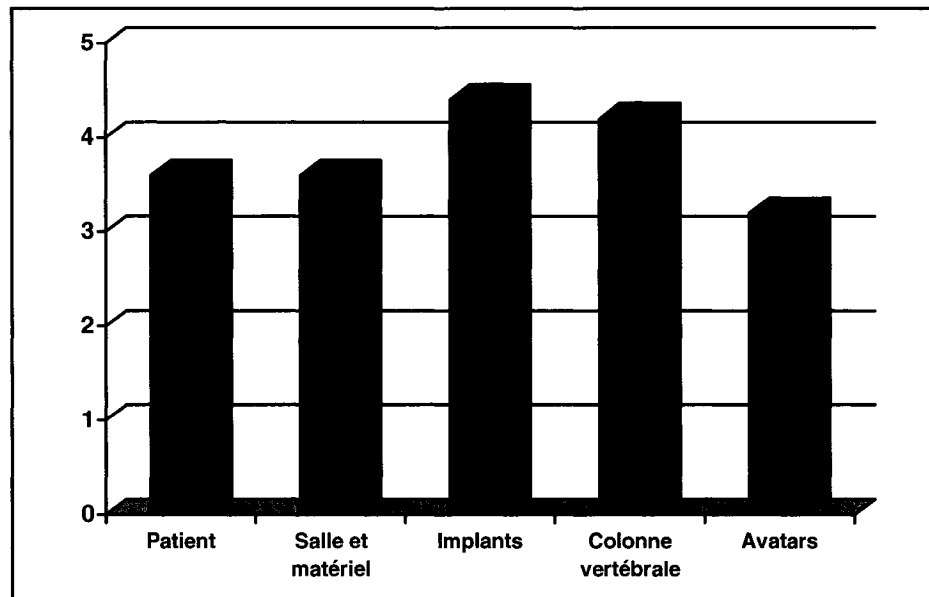
### 7.3.3 Résultats des tests

Les résultats des tests proviennent de deux sources différentes : les formulaires d'évaluation et les commentaires recueillis de vive voix. Les questionnaires ont été conçus, remis aux participants et recueillis par le concepteur. Le fait que la remise n'ait pas été confidentielle mérite d'être soulignée dans le sens où elle pourrait avoir influencé les résultats. Puisqu'une option de remise confidentielle a été proposée et qu'elle a été rejetée par les participants, nous pouvons supposer malgré tout que cette façon de faire n'aura pas trop influencé les résultats.

Le questionnaire distribué aux participants est présenté en annexe VIII. Il évalue le réalisme, l'interface immersive, l'interface non-immersive, le sentiment d'immersion, les tâches, l'aspect collaboratif et l'utilité future du prototype développé. Chaque catégorie est séparée en thèmes plus précis pouvant être évalués sur une échelle de 1 à 5.

#### **Réalisme**

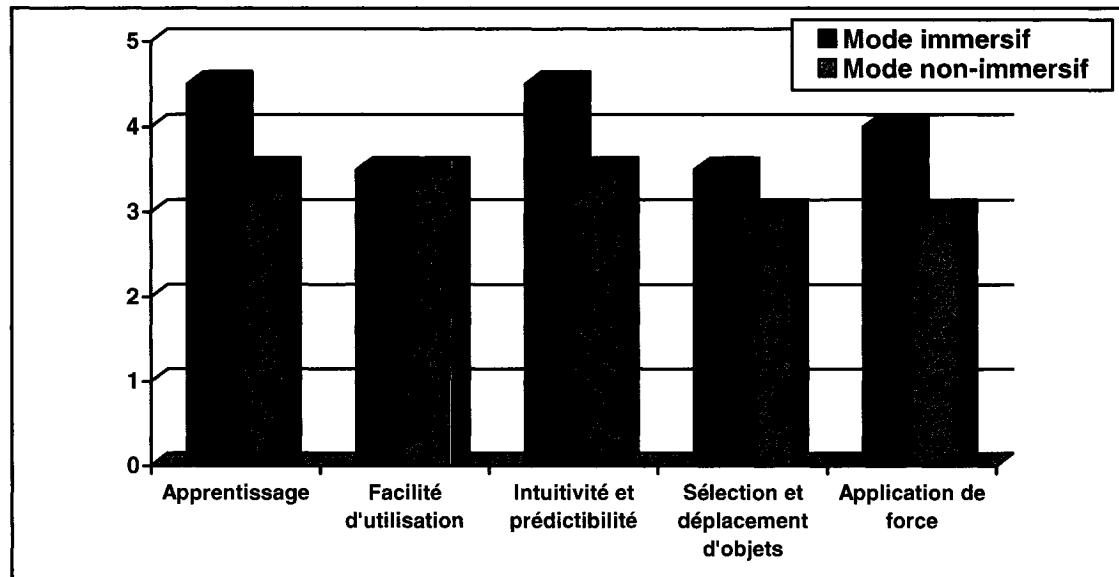
La figure 7.1 présente la moyenne des évaluations des cinq participants concernant le réalisme de la salle de chirurgie virtuelle et des objets en faisant partie. Leur évaluation est généralement positive, mais suggère une amélioration des avatars. Les commentaires recueillis proposent l'utilisation de vidéo faciale afin d'augmenter leur réalisme, mais mentionnent également la faible priorité de cet aspect.



**Figure 7.1** : Résultats du formulaire d'évaluation : réalisme géométrique.

### **Interface immersive et non-immersive**

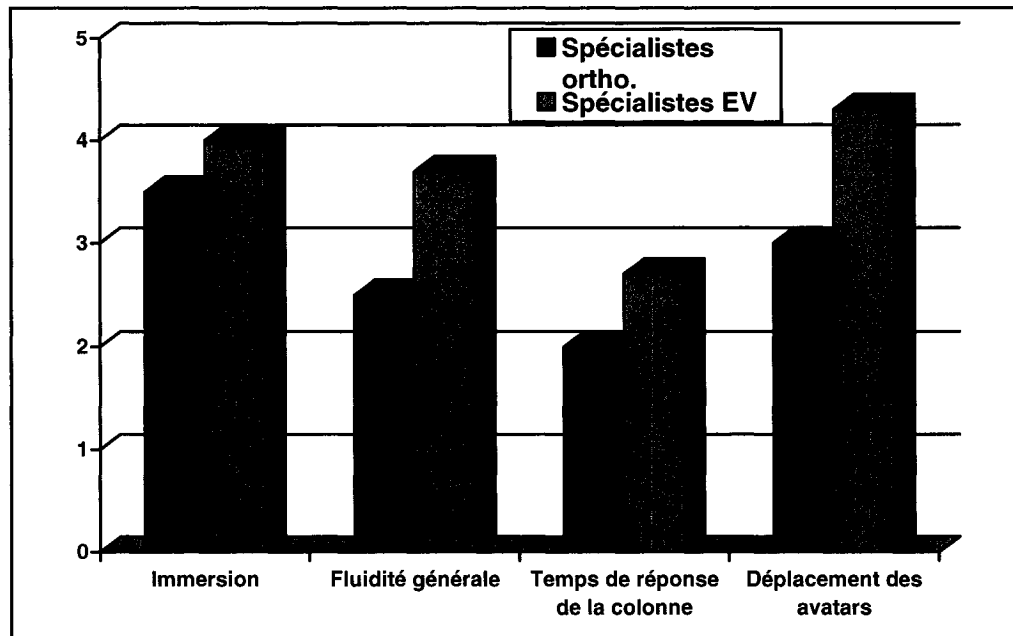
La figure 7.2 compare les résultats de l'évaluation de l'interface immersive à ceux de l'interface non-immersive. La tendance générale semble prouver l'hypothèse initiale selon laquelle il est plus facile d'interagir avec la simulation via l'utilisation d'un environnement de réalité virtuelle. Les commentaires recueillis mentionnent qu'il est parfois difficile de sélectionner un implant sur le plateau étant donné la haute concentration d'objets.



**Figure 7.2 :** Résultats du formulaire d'évaluation : interfaces.

### Immersion

La figure 7.3 présente les résultats de l'évaluation du sentiment d'immersion, du temps de réponse et de la fluidité générale de la simulation, du rachis scoliotique et des avatars. La séparation des résultats montre un grand écart entre les évaluations des deux groupes. L'aspect subjectif des thèmes évalués et un niveau d'expérience différent avec la RV peut potentiellement expliquer cette séparation. La faible évaluation du temps de réponse du rachis mérite par contre d'être soulignée. Cette évaluation subjective coïncide avec les observations et les tests présentés dans la section 6.3.11.

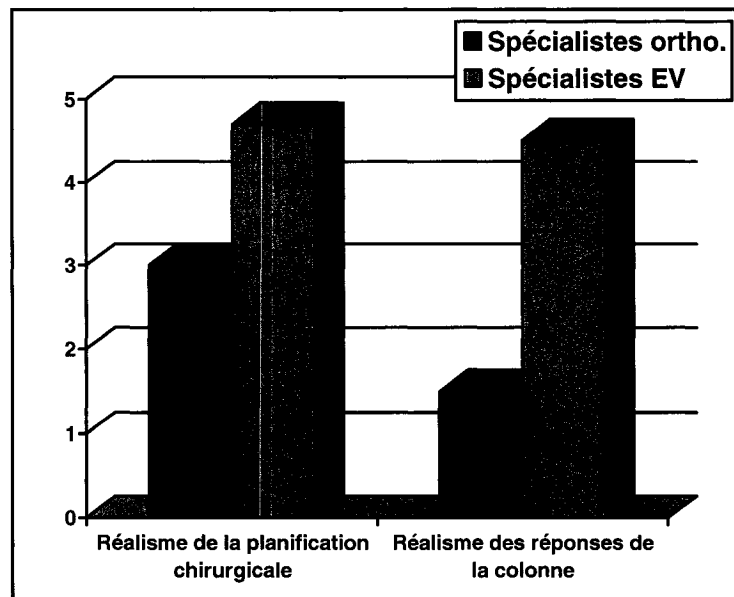


**Figure 7.3 :** Résultats du formulaire d'évaluation : immersion.

### Réalisme des tâches chirurgicales

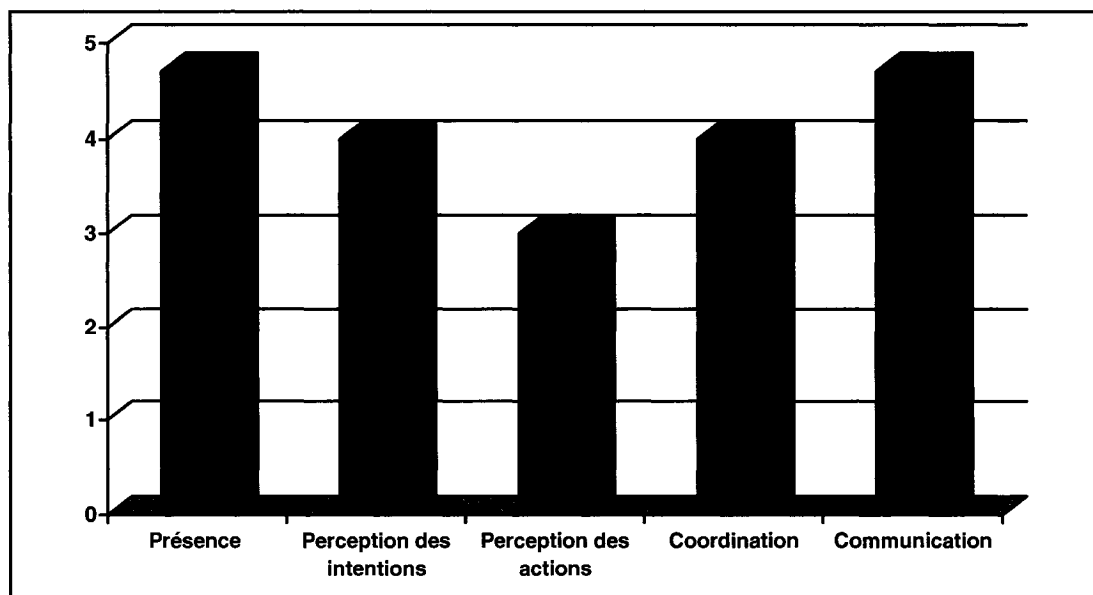
La figure 7.4 montre les résultats de l'évaluation du réalisme des tâches chirurgicales. Une séparation entre les groupes est encore proposée afin de souligner le grand écart entre leurs évaluations. Le fait que les spécialistes en EV ont une faible connaissance de la chirurgie scoliotique peut expliquer cette séparation.

Il est important de remarquer la faible évaluation attribuée par les spécialistes en orthopédie. Le peu de réalisme de la planification chirurgicale peut être associé aux commentaires reçus à propos de l'impossibilité d'effectuer l'éventail des procédures d'intervention. La faiblesse de l'évaluation du réalisme des réponses de la colonne est un peu plus difficile à interpréter étant donné l'utilisation du moteur biomécanique de S3. Quoique imparfaits, de nombreux tests ont démontré le réalisme des réponses biomécaniques de S3. Aucun commentaire à ce sujet n'a été formulé. L'hypothèse suivante semble par contre plausible: la longueur, la variabilité des délais et les situations où aucune solution n'est trouvée, par le module S3, rend irréaliste le comportement dynamique du rachis.



**Figure 7.4 :** Résultats du formulaire d'évaluation : réalisme des tâches chirurgicales.

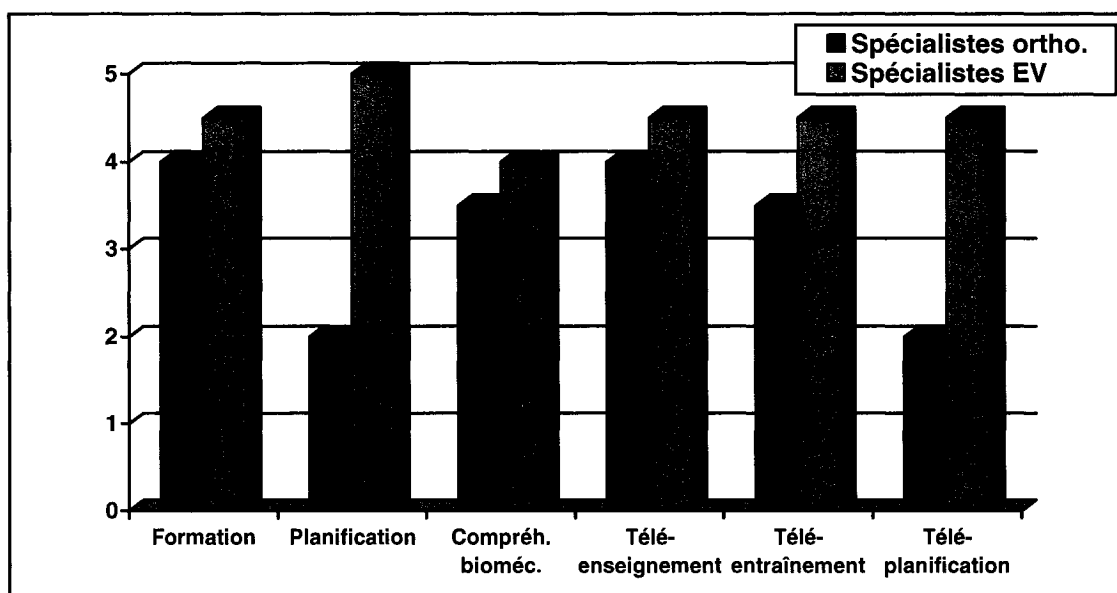
La figure 7.5 montre les résultats globaux de l'évaluation de l'aspect collaboratif du prototype. Ces résultats dressent un portrait fort positif des préoccupations au cœur de cette recherche. La perception des actions des utilisateurs mérite malgré tout d'être améliorée. Certains commentaires mentionnent qu'une visibilité des sélections et des déplacements d'objets des autres participants améliorerait beaucoup cet aspect.



**Figure 7.5 :** Résultats du formulaire d'évaluation : collaboration.

La figure 7.6 présente les résultats évaluant l'utilité future de ce type de simulateur pour les médecins résidents et les chirurgiens orthopédistes. Les résultats des deux groupes sont présentés afin d'obtenir un meilleur aperçu de l'évaluation des spécialistes en orthopédie qui sont, évidemment, les mieux placés pour juger cet aspect.

Les commentaires reçus et les évaluations des spécialistes en orthopédie confirment la forte utilité qu'aurait ce type de simulateur pour la formation générale des médecins résidents, la compréhension de la biomécanique du rachis scoliotique, le télé-enseignement et le télé-entraînement à l'instrumentation du rachis scoliotique. Ils jugent par contre qu'un tel outil ne serait pas l'idéal pour une planification ou une télé-planification chirurgicale cas par cas. Un outil comme S3, simple et rapide à utiliser, serait plus approprié pour effectuer ce type de tâche.



**Figure 7.6 :** Résultats du formulaire d'évaluation : utilité.



## CHAPITRE 8 : DISCUSSION

À la lumière des chapitres précédents, qui présentaient l'architecture globale du système, le serveur biomécanique, le serveur de téléprésence, le client de simulation et les tests usagers, ce chapitre propose une description des limites et des bénéfices du simulateur collaboratif développé pour cette étude. Les améliorations pouvant être apportées au prototype seront ensuite présentées ainsi que des avenues de recherches futures.

### 8.1 Limites et bénéfices

#### 8.1.1 Serveur biomécanique d'instrumentation collaborative du rachis scoliotique

L'absence des manœuvres de compression/distraktion et de dérotation directe des vertèbres nuit à la simulation d'une session d'instrumentation réaliste. Cette lacune, conséquente à la grande envergure du système à développer, n'est pas trop dramatique puisqu'il s'agit avant tout d'une étude de faisabilité. Implanter les deux manœuvres manquantes à partir de l'architecture établie est néanmoins une tâche facile étant donné l'état actuel du système. Les manœuvres implantées procurent malgré tout un bon aperçu des possibilités offertes et prouvent qu'il est possible d'intégrer le module de S3 dans un environnement d'entraînement en réalité virtuelle. Le serveur développé a également prouvé qu'il est possible d'adapter le module de S3 pour une utilisation collaborative.

La plus grande limite du serveur provient de la longueur et de la grande variabilité des délais associés aux requêtes d'application de manœuvres. Le résultat de cette limitation est que l'expérience procurée est amoindrie par rapport à celle obtenue lors d'une vraie chirurgie. Une vraie colonne vertébrale réagit en temps réel aux forces appliquées par le chirurgien et non d'une manière saccadée. Ce problème s'en trouverait déjà beaucoup diminué si l'ordinateur hébergeant le serveur était plus performant.

Une modification du code source du module S3 pourrait aussi améliorer cet aspect. L'objectif serait de procurer un accès à une ou des fonctions permettant de choisir le niveau de précision désiré quant au comportement biomécanique du rachis.

Le délai d'attente associé aux traitements des requêtes où le module S3 ne trouve pas de solution peut parfois être très long. Ces échecs sont souvent liés à la soumission de requêtes irréalistes d'un point de vue chirurgical. L'attachement d'une tige est fortement sujet à ce problème. Le fait que les manœuvres subséquentes sont grandement affectées par l'irréalisme d'une tige rend ce problème encore plus important. Une solution pourrait peut-être diminuer le nombre d'occurrences où cette situation se produit : la création d'un module de validation du réalisme d'une tige.

Les bénéfices du serveur biomécanique peuvent être résumés ainsi :

- Une représentation très réaliste du comportement biomécanique du rachis, grâce à l'intégration du module S3, lors de l'application de plusieurs manœuvres chirurgicales dans un environnement virtuel 3D;
- Un accès collaboratif à l'état du rachis scoliotique d'un patient spécifique et la possibilité de soumettre des requêtes d'instrumentation d'une manière concurrente ;
- Un serveur biomécanique dédié qui n'entretient aucune dépendance avec le design propre aux clients ;

### **8.1.2 Serveur de téléprésence multi-usagers**

Le serveur de téléprésence facilite la collaboration entre les participants lors d'une session d'entraînement chirurgical multi-usagers. Il augmente le sentiment de présence inter-participants grâce au contrôle qu'il exerce sur les avatars. Il améliore également la perception des intentions et la coordination inter-collaborateurs. Sa conception inhérente

et le fait qu'il soit multiplateforme implique qu'il peut facilement être réutilisé pour d'autres types d'EVC.

Il possède néanmoins la limite suivante : ils ne propage pas les états et les déplacements des objets faisant partie de la salle de chirurgie virtuelle. Mentionnons également que le réalisme des avatars pourrait être amélioré. La projection de vidéo sur les modèles des avatars pourrait probablement améliorer cet aspect. Il faut par contre remarquer que les avatars ne représentent que la position de la tête et de la souris 3D de l'utilisateur. Cette représentation statique, liée aux dispositifs utilisés, n'est pas fidèle aux mouvements effectués lors d'une vraie chirurgie. L'utilisation de gants de réalité virtuelle offrirait déjà une interface plus naturelle et impliquerait la génération d'avatars plus réalistes. Utiliser des avatars vidéos complets, plutôt que des représentations purement géométriques, offrirait également une vue beaucoup plus réaliste.

### **8.1.3 Client de simulation de chirurgie scoliotique en réalité virtuelle**

Le client de simulation est indirectement affecté par les limitations du serveur biomécanique (délais des requêtes, manœuvres non implantées). Les limites suivantes le concernent directement :

- Il peut être ardu, lors d'une première expérience, de sélectionner un implant du plateau à outils étant donné la haute concentration des objets présents ;
- La version non-immersive offre une interface de contrôle peu efficace étant donné la nature spatiale des actions à accomplir ;
- Une seule souris 3D peut être utilisée pour interagir avec la simulation. Une chirurgie scoliotique nécessite l'utilisation de ses deux mains et la souris, en plus d'être encombrante, n'est pas très précise (précision du système de repérage) ;

Les bénéfices suivants méritent d'être mentionnés :

- Il offre un environnement virtuel réaliste pour une simulation individuelle ou collaborative de l'instrumentation du rachis scoliotique ;
- Il offre une flexibilité d'utilisation grâce à son interface pour ordinateur conventionnel ;
- L'expérience procurée lors de l'utilisation en mode immersif est généralement semblable à celle d'une vraie chirurgie. Les gestes à accomplir lors de la procédure d'instrumentation se rapprochent de ceux qu'il faut accomplir lors d'une vraie chirurgie ;
- Il permet de s'entraîner seul ou en groupe, avant la réelle chirurgie, sur une modélisation du rachis scoliotique d'un patient spécifique.
- Les méthodes d'interaction proposées pour le mode immersif sont faciles à apprendre et utiliser, intuitives et prédictibles.
- Il est multiplateforme.

#### **8.1.4 L'ensemble du système**

Les évaluations des orthopédistes révèlent que le système développé ne serait pas idéal pour une planification ou une télé-planification cas par cas. La complexité inhérente au système de réalité virtuelle utilisé, la CAVE, est probablement liée à la faiblesse de cette évaluation. Il s'agit après tout d'un système nécessitant un grand espace, plusieurs ordinateurs et dispositifs spécialisés qui peuvent parfois être encombrants. Plusieurs recherches sont par contre en cours afin de rendre cette technologie plus accessible, moins intrusive et complexe. L'utilisation éventuelle de simples lunettes ou même de lentilles de contact, jumelés à des systèmes de repérage non intrusifs, contribuera peut-être à rendre cette technologie plus accessible.

L'évaluation des orthopédistes a par contre révélé qu'un tel système pourrait être utile pour :

- La formation générale des médecins résidents ;
- La compréhension de la biomécanique du rachis scoliotique ;
- La tenue de sessions de télé-enseignement à la chirurgie d'instrumentation du rachis scoliotique.
- La tenue de sessions de télé-entraînement à la chirurgie d'instrumentation collaborative du rachis scoliotique

Utiliser ce type de simulateur pourrait, en plus de réduire les coût associés aux méthodes d'entraînement actuels (section 1.1.3), faciliter, améliorer et accélérer le processus d'apprentissage de la chirurgie scoliotique. Il pourrait également être utilisé pour le perfectionnement des compétences opératoires des chirurgiens plus expérimentés. Il faut par contre remarquer qu'un tel outil est avant tout complémentaire : rien ne remplacera l'expérience gagnée lors de vraie chirurgies. Le simulateur offre néanmoins la possibilité de simuler des cas complexes et de tester plusieurs stratégies d'intervention pour un patient donné.

## **8.2 Avenues de recherches futures**

Lors de la poursuite de ce projet, il sera important de mieux définir le besoin auquel répondra le simulateur. Créer un simulateur pouvant être utilisé efficacement par la communauté médicale dans l'objectif global de permettre la formation générale des médecins résidents, la planification chirurgicale, la compréhension biomécanique du rachis, le télé-enseignement, le télé-entraînement et la télé-planification est une tâche ardue, voire impossible étant donné l'état d'avancement actuel. Les besoins auxquels devront répondre le prochain simulateur devront donc être mieux définis via une collaboration plus rapprochée entre concepteurs et orthopédistes.

### **8.2.1 Adaptation avancée du modèle biomécanique**

La distinction entre la raison d'être actuelle de S3 et du prototype développé peut être considérée comme une source de problèmes. S3 est un outil de planification se voulant rapide et facile à utiliser qui offre une interface d'interaction statique plutôt limitée. D'un autre côté, CVrS3 se veut un système d'entraînement qui simule la majorité des actions possibles en cours de chirurgie dans le but de créer une expérience se rapprochant le plus possible de la réalité. Il n'est donc pas conçu dans la même optique et n'accorde pas d'importance à la rapidité d'utilisation puisqu'il veut imiter le plus possible le déroulement normal d'une vraie chirurgie. Il serait donc intéressant de pousser encore plus loin l'adaptation du module S3 en fonction des besoins importants à une utilisation en réalité virtuelle.

Dans S3, pour attacher une tige à un ensemble d'implants il suffit d'avoir préalablement courbé la tige, de la déplacer, de choisir les implants concernés et l'ordre d'attachement. Le résultat de l'attachement est automatiquement visible après un court moment. Il serait intéressant de permettre à l'utilisateur d'appliquer une force sur le rachis et sur la tige afin qu'il puisse l'attacher de ses propres mains aux implants.

L'idée d'offrir plus de liberté quant aux actions possibles, qu'elles soient plus ou moins pertinentes, inutiles ou même dangereuses, rendrait l'expérience virtuelle beaucoup plus semblable à celle obtenue lors d'une vraie chirurgie. L'utilisateur pourrait par exemple appliquer une force sur un implant dans le seul but de mieux percevoir le comportement biomécanique du rachis.

### **8.2.2 Amélioration de l'interface non-immersive**

Les tests usagers ont révélé à quel point il était difficile d'interagir avec la simulation et d'effectuer des actions de nature spatiale à l'aide du clavier et de la souris. L'utilisation

d'une petite unité de contrôle 3D (associée à la souris 3D) repérée dans un volume limité par un système de repérage à faible coût pourrait probablement améliorer la facilité de l'interaction.

Une alternative : utiliser un petit système haptique offrant une liberté de mouvement suffisante pour interagir avec la simulation.

### **8.2.3 Utilisation de gants de réalité virtuelle**

Le simulateur actuel est conçu pour l'utilisation d'une seule souris 3D. Le fait qu'une chirurgie scoliotique nécessite l'utilisation des deux mains a rendu cet aspect problématique. L'utilisation d'une paire de gants de réalité virtuelle permettrait donc d'augmenter le réalisme et la facilité d'utilisation du simulateur. Un gant est généralement moins encombrant qu'une souris 3D.

Ce choix de dispositif nécessiterait une adaptation des méthodes d'interaction implantées pour le prototype développé dans le cadre de cette étude.

### **8.2.4 Systèmes haptiques**

L'utilisation de systèmes haptiques et de gants de réalité virtuelle augmenterait énormément le réalisme de la simulation. Deux systèmes haptiques (un pour chaque main) permettraient de ressentir le retour d'effort associé à l'application des manœuvres correctrices telles qu'exécutées dans une vraie chirurgie. L'implantation d'un guidage des actions maître/élève(s), à l'aide de plusieurs systèmes haptiques interconnectés, pourrait d'ailleurs être fort utile pour la formation des médecins résidents. Un étudiant pourrait ainsi ressentir les forces appliquées, en cours de manœuvre, par un chirurgien expérimenté.

### **8.2.5 Reconnaissance vocale**

Les tests usagers ont démontré qu'il pouvait parfois être difficile de sélectionner des objets sur le plateau à outils. Un système de reconnaissance vocale pourrait grandement simplifier la sélection des implants et des outils. Cette sélection automatique via la voix simule également les circonstances où le chirurgien demande à une infirmière de lui donner un outil.



## CONCLUSION

Cette étude a prouvé la faisabilité du développement d'un simulateur collaboratif de chirurgie scoliotique en réalité virtuelle intégrant la plus récente version du module S3 de modélisation biomécanique du rachis scoliotique. Elle prouve qu'une telle modélisation, isolée sur un serveur dédié, peut être accédée d'une manière concurrente pour gérer et rendre accessible l'état du rachis lors de sessions d'entraînement collaboratif, formés de participants géographiquement éloignés, effectuées à l'intérieur d'une salle de chirurgie virtuelle. Elle prouve également la supériorité des systèmes immersifs de réalité virtuelle par rapport aux interfaces conventionnelles, avec écran, clavier et souris standard, en ce qui a trait au réalisme de la simulation et à la facilité d'interaction. Les tests usagers effectués, bien que de petite envergure, mettent en évidence l'utilité que pourrait avoir ce type de simulateur pour la formation générale des médecins résidents, la compréhension de la biomécanique du rachis scoliotique, le télé-enseignement et le télé-entraînement à la chirurgie d'instrumentation du rachis scoliotique.

Ce projet de recherche a permis, plus précisément, le développement d'un serveur biomécanique collaboratif, d'un serveur de téléprésence multi-usagers et d'un client de simulation chirurgicale en réalité virtuelle. La possibilité d'utiliser le client de simulation à partir d'un ordinateur conventionnel diminue la barrière d'entrée et offre plus de flexibilité quant aux scénarios d'utilisation possibles. Ces entités, et l'architecture réseautique inhérente, peuvent servir de bases à la création d'un simulateur plus complet pouvant, éventuellement, servir la communauté médicale.

Bien que l'avenir du simulateur semble prometteur, un long processus de recherche et de développement demeure à accomplir avant l'obtention d'un système concrètement utilisable. L'utilisation de gants de réalité virtuelle et de systèmes haptiques est d'ailleurs primordiale à l'obtention d'une simulation réaliste. L'adaptation du moteur de modélisation biomécanique est également nécessaire. Le dynamisme et la fluidité du

comportement de la colonne vertébrale, lors de l'application de forces correctrices, sont indispensables à la création d'une expérience se rapprochant le plus possible d'une réelle intervention.

Ce travail est un petit pas pour l'homme. Il ouvre néanmoins la voie à ce qui sera, peut-être, un grand pas pour l'humanité...

## Références

- [**Ai et al. 2000**] Ai, Z., Gupta, B.K., Rasmussen, M., Lin, Y.J., Dech, F., Panko, W., Silverstein, J.C.  
*Simulation of Eye Diseases in a Virtual Environment.*  
 Hawaii International Conference on System Sciences (HICSS-33), Janvier 2000.
- [**Aggarwal et al. 2004**] Aggarwal, R., Hance, J., Darzi, A.  
*Surgical education and training in the new millennium.*  
 Surgical Endoscopy, 18(10), pp 1409–1410, Springer New York, Octobre 2004.
- [**Porter et al. 2004**] Porter, S.C., Cai, Z., Gribbons, W., Goldmann, D.A., Kohane, I.S.  
*The Asthma Kiosk: A Patient-centered Technology for Collaborative Decision Support in the Emergency Department.*  
 J Am Med Inform Assoc, 11(6), pp 458–467, Novembre-Décembre 2004.
- [**Aubin et al. 2006, 1**] Aubin, C.É., Labelle, H., Lenke, L., Lowe, T., Emans, J., Sucato, D., Kuklo, T.  
*Variability of Spinal Instrumentation Configuration in Adolescent Idiopathic Scoliosis.*  
 Eur Spine J., pp 1-8, Février 2006.
- [**Aubin et al. 2006, 2**] Aubin, C.É., Labelle, H., Chevretils, C., Clin, J.  
*Pre-Operative Planning Simulator for Spinal Deformity Instrumentation Surgeries.*  
 En préparation.
- [**Blyth et al. 2006**] Blyth, P., Anderson, I.A., Stott, N.S.  
*Virtual Reality Simulators in Orthopedic Surgery: What Do the Surgeons Think?*  
 Journal of Surgical Research, 131, pp 133-139, 2006.
- [**Carlin et al. 1997**] Carlin, A.S., Hoffman, H.G., Weghorst, S.  
*Virtual reality and tactile augmentation in the treatment of spider phobia: a case report.*  
 Behav. Res. Ther., 35(2), pp 153-158, Février 1997.
- [**Caudell et al. 2003**] Caudell, T.P., Summers, K.L., IV, J.H., Hakamata, T., Mowafi, M., Jacobs, J., Lozanoff, B.K., Lozanoff, S., Wilks, D., Keep, M.F., Saiki, S., Alverson, D.  
*Virtual Patient Simulator for Distributed Collaborative Medical Education.*  
 The Anatomical Record, Part B: New Anat., 270B, pp 23-29, 2003.
- [**Choi et al. 2005**] Choi, Y.J., Choi, S.M., Rhee, S.M., Kim, M.H.  
*Collaborative and Immersive Medical Education in a Virtual Workbench Environment.*  
 KES 2005, pp 1210-1217, 2005.

**[Kahn 2006]** Kahn C.E. Jr.

*About CHORUS - Collaborative Hypertext of Radiology.*

Novembre 2006, ([chorus.rad.mcw.edu/about-CHORUS.html](http://chorus.rad.mcw.edu/about-CHORUS.html)) – Consulté le 10 Janvier 2007.

**[Scoliosis Association]** Scoliosis Association, Inc.

*Types of scoliosis.*

([www.scoliosis-assoc.org](http://www.scoliosis-assoc.org)) – Consulté le 10 Mai 2006.

**[El-Far et al. 2005]** El-Far, N.R., Nourian, S., Zhou, J., Hamam, A., Shen, X., Georganas, N.D.

*A Cataract Tele-Surgery Training Application in a Hapto-Visual Collaborative Environment Running over the CANARIE Photonic Network.*

HAVE 2005, IEEE International Workshop on Haptic Audio Visual Environments and their Applications, pp 29-32, Octobre 2005.

**[Gardner-Morse et al. 1990]** Gardner-Morse, M.G., Laible, J.P., Stokes, I.A.

*Incorporation of spinal flexibility measurements into finite element analysis.*

J. Biomech. Eng., 112, pp 481-493, 1990.

**[Grantcharov et al. 2004]** Grantcharov, T. P., Kristiansen, V. B., Bendix, J., Bardram, L., Rosenberg, J., Funch-Jensen, P.

*Randomized clinical trial of virtual reality simulation for laparoscopic skills training.*

Br. J. Surg., 91, 146, 2004.

**[Johnsen et al. 2005]** Johnsen, K., Dickerson, R., Raij, A., Lok, B., Jackson, J., Shin, M., Hernandez, J., Stevens, A., Lind, S.

*Experience in Using Immersive Virtual Characters to Educate Medical Communication Skills.*

Proceedings of IEEE Virtual Reality 2005.

**[Johnsen et al. 2006]** Johnsen, K., Dickerson, R., Raij, A., Harrison, C., Lok, B.

*Evolving an Immersive Medical Communications Skills Trainer.*

Presence, 15(1), pp 1-14, Février 2006.

**[Kauff et Schreer 2002]** Kauff, P., Schreer, O.

*An Immersive 3D Video-Conferencing System Using Shared Virtual Team User Environments.*

CVE'02, pp 105-112, 2002.

**[Kizony et al. 2002]** Kizony, R., Katz, N., Weingarden, H., Weiss, L.

*Immersion without encumbrance: adapting a virtual reality system for the rehabilitation of individuals with stroke and spinal cord injury.*

4th International Conference on Disability, Virtual Reality & Assoc., pp 55-61, 2002.

**[Ku et al. 2003]** Ku, J., Cho, W., Kim, J.J., Peled, A., Wiederhold, B.K., Wiederhold, M.D., Kim, I.Y., Lee, J.H., Kim, S.I.

*A virtual environment for investigating schizophrenic patients' characteristics: assessment of cognitive and navigation ability.*

Cyberpsychol. Behav., 6(4), pp 397-404, Août 2003.

**[Lamade et al. 2000]** Lamade, W., Glombizza, G., Fischer, L., Chiu, P., Cardenas, C.E., Thorn, M.

*The Impact of 3-Dimensional Reconstructions on Operation Planning in Liver Surgery.*

Arch. Surg., 135, pp 1256-1261, 2000.

**[Lin et al. 2002]** Lin, C.Y., Chen, D.T., Loftin, R.B., Chen, J., Leiss, E.L.

*Interacting with Visible Human Data Using an ImmersaDesk.*

Proceedings of IEEE Virtual Reality 2002.

**[Lonstein 1995]** Lonstein, J.E.

*Moe's Textbook of Scoliosis and Other Spinal Deformities.*

Philadelphia, W.B. Saunders, 1995.

**[Luce 2004]** Luce, Fabien

*Contribution à l'amélioration du réalisme d'un modèle multi-corps flexibles de chirurgie pour le traitement de la scoliose.*

Mémoire de maîtrise (École Polytechnique de Montréal), 2004.

**[Maithel et al. 2006]** Maithel, S., Sierra, R., Korndorffer, J., Neumann, P., Dawson, S., Callery, M., Jones, D., Scott, D.

*Construct and face validity of MIST-VR, Endotower, and CELTS. Are we ready for skills assessment using simulators?*

Surg. Endosc., 20, pp 104–112, 2006.

**[Mantovani 2003]** Mantovani, F.

*VR learning: Potential and Challenges for the Use of 3D Environments in Education and Training.*

Chapter 12 in *Towards Cyber-Psychology : Mind, Cognitions and Society in the Internet Age*, Amsterdam, IOS Press, 2001, 2002, 2003.

**[Megali et al. 2002]** Megali, G. , Tonet, O., Mazzoni, M., Dario, P., Vascellari, A., Marcacci, M.

*A New Tool for Surgical Training in Knee Arthroscopy.*

MICCAI 2002, LNCS 2489, pp 170–177, 2002.

**[Meiguins et al. 2004]** Meiguins, B.S., Meiguins, B.S., de Brito Garcia, M., de Souza Jr, R.D., Guedes, L.A.

*Collaborative Virtual Lab for Electronic Experience.*

Proceedings of the IASTED International Conference, pp 86-88, Février 2004.

**[Morris et al. 2004]** Morris, D., Sewell, C., Blevins, N., Barbagli, F., Salisbury, K.

*A Collaborative Virtual Environment for the Simulation of Temporal Bone Surgery.*

MICCAI 2004, LNCS 3217, pp 319–327, 2004.

**[NIAMS 2001]** National Institute of Arthritis and Musculoskeletal and Skin Diseases

*Questions and Answers about Scoliosis in Children and Adolescents.*

01-4862: pp 1-6, 2001.

**[Oden et Ripperger 1981]** Oden, J.T., Ripperger, E.A.

*Mechanics of Elastic Structures.*

Second Edition ed., McGraw-Hill, New-York, 351, 1981.

**[Ozell 2003]** Ozell, B.

*Environnements d'immersion.*

École Polytechnique de Montréal, Septembre 2003, ([www.polymtl.ca/rv/rv/](http://www.polymtl.ca/rv/rv/)) – Consulté le 5 Mai 2006.

**[Panjabi et al. 1976]** Panjabi, M.M., Brand, R.A Jr., White, A.A. 3rd

*Three-dimensional flexibility and stiffness properties of the human thoracic spine.*

J. Biomech., 9, pp 185-192, 1976.

**[Perdriolle 1979]** Perdriolle, R.

*La scoliose. Son étude tridimensionnelle.*

Maloine S.A. Editeur, 1979.

**[Petit 2001]** Petit, Yvan

*Reconstruction 3D et modélisation géométrique du tronc scoliotique. Description de l'approche proposée.*

Hôpital Sainte-Justine, École Polytechnique, Juin 2001.

**[Petit et al. 2002]** Petit, Y., Aubin, C.É., Labelle, H.

*Three-dimensional imaging for the surgical treatment of idiopathic scoliosis in adolescents.*

Surgical Biology for the Clinician, pp 453-458, Décembre 2002.

**[Petit et al. 2004]** Petit, Y., Aubin, C.É., Labelle, H.

*Patient-Specific Mechanical Properties of a Flexible Multi-Body Model of the Scoliotic Spine.*

Medical & Biological Engineering & Computing, 42(1), pp 55-60, 2004.

**[Peuchot et al. 1995]** Peuchot, B., Tanguy, A., Eude, M.

*Virtual Reality as an Operative Tool during Scoliosis Surgery.*

Proceedings of CVRMed'95, Nice, France, 1995.

**[Plouznikoff 2004]** Plouznikoff, Alexandre

*Simulateur de chirurgie d'instrumentation du rachis scoliotique en réalité virtuelle.*

Mémoire de maîtrise (Ecole Polytechnique de Montréal), 2004.

**[Rao et Turoff 2000]** Rao, G.R., Turoff, M.

*A hypermedia-based group decision support system to support collaborative medical decision-making.*

Decision Support Systems, 30(2), pp 187-216, Décembre 2000.

**[Rasmussen et al. 1998]** Rasmussen, M., Mason, T.P., Millman, A., Evenhouse, R., Sandin, D.

*The virtual temporal bone, a tele-immersive educational environment.*

Future Generation Computer Systems, 14, pp 125-130, 1998.

**[Rothbaum et al. 1999]** Rothbaum, B.A., Hodges, L., Alarcon, R., Ready, D., Shahar, F., Graap, K., Pair, J., Hebert, P., Gotz, D., Wills, B., Baltzell, D.

*Virtual Reality Exposure Therapy for PTSD Vietnam Veterans: A Case Study.*

Journal of Traumatic Stress, 12(2), pp 263-271, Avril 1999.

**[Schlusselberg et al. 1988]** Schlusselberg, D.S., Smith, W.K., Woodward, D.J., Parkey, R.W.

*Use of Computed Tomography for a 3-Dimensional Treatment Planning System.*

Comput. Med. Imaging Graph., 12, pp 25-32, 1988.

**[Scoliosis Association]** Scoliosis Association, Inc.

*Types of scoliosis.*

([www.scoliosis-assoc.org](http://www.scoliosis-assoc.org)) – Consulté le 10 Mai 2006.

**[Seymour et al. 2002]** Seymour, N. E., Gallagher, A. G., Roman, S. A., et al.

*Virtual reality training improves operating room performance: Results of a randomized, double-blinded study.*

Ann. Surg., 236, 458, discussion 463, 2002.

**[Smith 2001]** Smith, R.C.

*Shared Vision. General Motors proved fertile ground for implementing collaborative technologies for designing cars.*

Communications of the ACM, 44(12), pp 45-48, Décembre 2001.

**[SRS 2000]** Scoliosis Research Society

*In Depth Review of Scoliosis.*

Septembre 2000, ([www.srs.org](http://www.srs.org)) – Consulté le 11 Mai 2006.

**[Stylopoulos et al. 2003]** Stylopoulos, N., Cotin, S., Dawson, S., Ottensmeyer, M., Neumann, P., Bardsley, R., Russell, M., Jackson, P., Rattner, D.

*CELTS: A clinically-based Computer Enhanced Laparoscopic Training System.*

MMVR, 2003.

**[Sutton et al. 1997]** Sutton, C., McCloy, R., Middlebrook, A., Chater, P., Wilson, M., Stone, R.

*MIST VR. A laparoscopic surgery procedures trainer and evaluator.*

Stud. Health Technol. Inform., 39, pp 598-607, 1997.

**[Tanguy et Peuchot 2002]** Tanguy, A., Peuchot, B.

*Systèmes overview : applications et avenir.*

Chirurgie orthopédique assistée par ordinateur, pp 183-191, 2002.

**[Verdaasdonk et al. 2006]** Verdaasdonk, E.G.G., Stassen, L.P.S., Monteny, L.J., Dankelman, J.

*Validation of a new basic virtual reality simulator for training of basic endoscopic skills.*

Surg. Endosc., 20, pp 511-518, 2006.

**[Wiederhold et al. 2002]** Wiederhold, B.K., Jang, D.P., Gevirtz, R.G., Kim, S.I., Kim, I.Y., Wiederhold, M.D.

*The treatment of fear of flying: a controlled study of imaginal and virtual reality graded exposure therapy.*

IEEE Transactions on Information Technology in Biomedecine, 6(3), pp 218-223, Septembre 2002.

**[Wu et al. 1998]** Wu, J., Song, G., Yeh, C.P., Wyatt, K.

*DropAmpact Simulation and Test Validation of Telecommunication Products*

1998 InrerSociety Conference on Thermal Phenomena, pp 330-336, 1998.

**[Zajtchuk et Satava 1997]** Zajtchuk, R., Satava, R.M.

*Medical Applications of Virtual Reality.*

Communications of the ACM, 40(9), pp 63-64, 1997.





**Socket** est une classe provenant de la librairie C++ Sockets Library de laquelle dérivent d'autres classes qui implémentent différents protocoles internet et protocoles de plus haut niveau. La classe **TCPSocket**, qui implémente le protocole TCP et qui est utilisée par le serveur, est l'une des classes dérivées pouvant être utilisées. La classe **SocketHandler** joue le rôle de contrôleur, de gestionnaire d'événements et de répartiteur pour les classes qui dérivent de **Socket**. Elle gère donc les événements (lecture/écriture/exception) de la classe **BiomechServerInterface** et les lui reporte à l'aide de méthodes de *callback*. Les fonctions de *callback* **OnAccept()** et **OnLine(...)** sont appelées lors de l'acceptation d'une nouvelle connexion et lors de la réception d'un message formé d'une chaîne de caractères d'une seule ligne. Puisque chaque requête d'un client est exprimée selon ce format, la fonction **OnLine(...)** constitue le point d'entrée où leurs requêtes sont reçues, analysées et traitées par l'appel de la fonction appropriée

(**biomechServerInitAdams()**,  
**biomechServerAddImpAndSendInfos(...)**,  
**biomechServerAttachNewRodToAllImpsAndSendInfos(...)**, ...). Une nouvelle instance de la classe **BiomechServerInterface** est créée pour chaque nouvelle connexion avec un client. Les communications entre les clients et le serveur sont donc gérées à partir de *sockets* indépendants qui s'exécutent dans leurs propres *threads* afin d'assurer un traitement parallèle. L'ensemble des classes présentées correspond à la boîte *Interface réseau* de la figure 4.1.

La classe **SpineServer** correspond au module d'instrumentation collaborative de la figure 4.1. Elle est conçue selon le patron singleton afin que tous les clients accèdent à une seule et même copie de l'état du rachis scoliotique du patient. Les multiples instances de la classe **BiomechServerInterface** (une instance par client connecté) accèdent donc, via la fonction **Instance()**, à une seule et même instance de la classe **SpineServer**. Les données caractérisant les clients participant à la session d'instrumentation sont conservées dans cette classe. Les fonctions qu'elles offrent simplifient l'interaction avec **CAdamsManager** et s'adaptent aux méthodes d'interaction

développées pour le simulateur en réalité virtuelle. Elles s'occupent également d'exécuter la séquence d'appels nécessaires (de **CAdamsManager**) pour appliquer chaque type de manœuvre et pour récupérer/retourner les données modifiées suite à son application.

La classe **CAdamsManager** correspond au module de manœuvres chirurgicales de la figure 4.1. Cette classe, qui provient du code source du logiciel S3, mais qui a été légèrement modifiée, offre des fonctions d'initialisation du moteur biomécanique et d'instrumentation du rachis. Malgré le fait que son design soit adapté au type d'interaction (interface graphique avec menus) offert par S3, il demeure assez flexible pour une utilisation en réalité virtuelle. Des fonctions utilitaires ont été ajoutées et certaines fonctions ont subi des modifications (ex. : élimination de lignes de code non utilisées, meilleure portabilité) afin de simplifier cette adaptation. **CAdamsManager** constitue une porte d'entrée vers le module S3 qui modélise le comportement de la colonne lors de l'instrumentation.

Le module S3 est composé de quatre autres modules : **HSJAdams**, **HSJLisFiles**, **HSJMath** et **HSJAnalysis**. Le module **HSJAdams** est le plus important des quatre modules puisque c'est lui qui modélise le comportement de la colonne vertébrale. Les trois autres modules aident **HSJAdams** à effectuer cette tâche et/ou participent aux tâches de la classe **CAdamsManager**. Le code source de **HSJAdams**, qui a été développé au LMBCAO, est écrit à l'aide du SDK Adams afin d'offrir une interface entre le simulateur et le logiciel de modélisation biomécanique MSC Adams.

## ANNEXE II : Description des fonctions du module de manœuvres chirurgicales

Le tableau II.1 présente et donne une description des fonctions dédiées pour chaque manœuvre. Ces fonctions font partie de la classe CAdamsManager du serveur biomécanique.

**Tableau II.1** : Description des fonctions de manœuvres chirurgicales de CAdamsManager.

<b>Fonction :</b>	<code>tStatus addImplant(const tImplant&amp; inImplant, CAdamsImplant*&amp; inoutImplant);</code>
<b>Manœuvre :</b>	Insertion d'un implant à une position pré-calculée.
<b>Description :</b>	Cette fonction est appelée pour insérer un implant dans une vertèbre spécifique. Le paramètre <code>inImplant</code> caractérise l'implant à insérer. Il spécifie notamment le type d'implant et la position pré-calculée où devra être inséré l'implant. Le paramètre <code>inoutImplant</code> contient, lors du retour de la fonction, toutes les informations caractérisant le nouvel implant vertébral. Sa matrice de position et d'orientation est d'ailleurs accessible via ce paramètre.
<b>Fonction :</b>	<code>tStatus addRod(const tRod &amp;inRod, CAdamsRod *&amp;inoutRod);</code>
<b>Manœuvre :</b>	Création d'une tige et modélisation de sa courbure.
<b>Description :</b>	Cette fonction est appelée pour créer une tige. Le paramètre <code>inRod</code> spécifie les points de contrôle de la tige ainsi que d'autres attributs la caractérisant. L'objet <code>inoutRod</code> altéré peut ensuite être utilisé lors de l'attachement de la tige précédemment créée.

<b>Fonction :</b>	<code>tStatus updateImplant(const unsigned int &amp;inObjID,                           const CMatrix2 &amp;inPose);</code>
<b>Manœuvre :</b>	Insertion d'un implant à une position spécifique.
<b>Description :</b>	Cette fonction met à jour la position d'un implant déjà inséré. Le paramètre <code>inObjID</code> spécifie le numéro d'identification de l'implant concerné. Le paramètre <code>inPose</code> permet de spécifier une matrice de position et d'orientation qui n'a pas été pré-calculée. Lorsque cette fonction est appelée immédiatement après <code>addImplant(...)</code> , elle peut être considérée comme une manœuvre permettant l'insertion d'un implant à une position spécifique.
<b>Fonction :</b>	<code>tStatus removeAnyImplant(const unsigned int inObjID);</code>
<b>Manœuvre :</b>	Enlèvement d'un implant.
<b>Description :</b>	Cette fonction permet d'enlever un implant vertébral. Le paramètre <code>inObjID</code> spécifie le numéro d'identification de l'implant à enlever.
<b>Fonction :</b>	<code>tStatus attachRod(const tSpineSide inSide,                           const bool inIsHold,                           const vector&lt;unsigned int&gt; inImplantList,                           const int inRodID,                           const string inResourcePath,                           const string inPatientName,                           bool inFromFile = false);</code>
<b>Manœuvre :</b>	Attachement d'une tige.
<b>Description :</b>	Cette fonction effectue une manœuvre d'attachement de tige. <code>inSide</code> est le côté de la colonne pour l'attachement, <code>inIsHold</code> indique si la tige est maintenue, <code>inImplantList</code> est un vecteur des numéros d'identification des implants vertébraux à attacher, <code>inRodID</code> est le numéro d'identification de la tige à attacher, <code>inResourcePath</code> indique le chemin du répertoire des données du patient, <code>inPatientName</code> indique le nom du patient et <code>inFromFile</code> indique si la simulation doit être effectuée à partir d'un fichier préalablement généré.

<b>Fonction :</b>	<code>tStatus detachRod(const int inRodID,                     const string inResourcePath,                     const string inPatientName,                     bool inFromFile = false);</code>
<b>Manœuvre :</b>	Détachement d'une tige.
<b>Description :</b>	Cette fonction détache une tige préalablement attachée. <code>inRodID</code> est le numéro d'identification de la tige à détacher, <code>inResourcePath</code> indique le chemin du répertoire des données du patient, <code>inPatientName</code> indique le nom du patient et <code>inFromFile</code> indique si la simulation doit être effectuée à partir d'un fichier préalablement généré.
<b>Fonction :</b>	<code>tStatus rotateRod(const int inRodID,                     const int inAngle,                     const tRotationDirection inDirection,                     const string inResourcePath,                     const string inPatientName,                     bool inFromFile = false);</code>
<b>Manœuvre :</b>	Rotation d'une tige.
<b>Description :</b>	Cette fonction effectue une manœuvre de rotation de tige. La tige doit avoir été préalablement attachée. <code>inRodID</code> est le numéro d'identification de la tige, <code>inAngle</code> est l'angle de rotation désiré, <code>inDirection</code> est la direction de rotation, <code>inResourcePath</code> indique le chemin du répertoire des données du patient, <code>inPatientName</code> indique le nom du patient et <code>inFromFile</code> indique si la simulation doit être effectuée à partir d'un fichier préalablement généré.
<b>Fonction :</b>	<code>tStatus lockUpRod(const int inRodID,                     const vector&lt;tImplant&gt; &amp;inImplantList,                     const string inResourcePath,                     const string inPatientName,                     bool inFromFile = false);</code>
<b>Manœuvre :</b>	Serrage d'une tige.
<b>Description :</b>	Cette fonction permet de serrer les écrous des implants spécifiés. La tige doit avoir été préalablement attachée aux implants spécifiés. <code>inRodID</code> est le numéro d'identification de la tige, <code>inImplantList</code> est un vecteur des numéros d'identification des implants vertébraux à serrer, <code>inResourcePath</code> indique le chemin du répertoire des données du patient, <code>inPatientName</code> indique le nom du patient et <code>inFromFile</code> indique si la simulation doit être effectuée à partir d'un fichier préalablement généré.

<b>Fonction :</b>	<pre> tStatus contractDistract(const unsigned int inImplant1,                         const unsigned int inImplant2,                         const REAL inFinalDistance,                         const string inResourcePath,                         const string inPatientName,                         bool inFromFile = false); </pre>
<b>Manœuvre :</b>	Compression ou distraction de vertèbres.
<b>Description :</b>	<p>Cette fonction permet de compresser (rapprocher) ou de distraire (éloigner) deux implants vertébraux le long d'une tige préalablement attachée. <b>inImplant1</b> est le numéro d'identification du premier implant, <b>inImplant2</b> est le numéro d'identification du deuxième implant, <b>inFinalDistance</b> est la distance finale désirée entre les deux implants, <b>inResourcePath</b> indique le chemin du répertoire des données du patient, <b>inPatientName</b> indique le nom du patient et <b>inFromFile</b> indique si la simulation doit être effectuée à partir d'un fichier préalablement généré.</p>
<b>Fonction :</b>	<pre> tStatus vertebraDerotation(const vector&lt;unsigned int&gt;                         inImplantList,                         const tAnatomy inVertebra,                         const int inRodID,                         const bool inIsHold,                         const REAL inFinalRotation,                         const string inResourcePath,                         const string inPatientName,                         bool inFromFile = false); </pre>
<b>Manœuvre :</b>	Dérotation vertébrale directe (DVD).
<b>Description :</b>	<p>Cette fonction effectue une manœuvre de dérotation vertébrale directe. <b>inImplantList</b> est un vecteur des numéros d'identification des implants vertébraux à utiliser pour l'application d'une force de dérotation, <b>inVertebra</b> est la vertèbre pour laquelle on effectue une dérotation, <b>inRodID</b> est le numéro d'identification de la tige attachée aux implants, <b>inIsHold</b> indique si la tige est maintenue lors de la dérotation, <b>inFinalRotation</b> indique l'angle final de dérotation désiré pour la vertèbre spécifiée (en rapport à son orientation initiale), <b>inResourcePath</b> indique le chemin du répertoire des données du patient, <b>inPatientName</b> indique le nom du patient et <b>inFromFile</b> indique si la simulation doit être effectuée à partir d'un fichier préalablement généré.</p>

## ANNEXE III : Structures des messages réseaux envoyés ou reçus par le serveur biomécanique

### Requête d'initialisation du moteur biomécanique

Cette requête est exprimée via l'étiquette suivante :

```
INIT_ADAMS
```

Le message de confirmation suivant est envoyé au client si le moteur a réussi son initialisation ou s'il était déjà initialisé :

```
ADAMS_READY
```

### Requête d'obtention des marqueurs des points d'insertion pré-calculés

Cette requête est exprimée via l'étiquette suivante :

```
GET_MARKERS
```

La réponse à cette requête utilise la structure suivante :

```
MARKERS: m0,x m0,y m0,z m1,x m1,y m1,z . . . m92,x m92,y m92,z
```

où chaque trio  $m_{i,x}$   $m_{i,y}$   $m_{i,z}$  correspond aux coordonnées x, y, z du i<sup>e</sup> point d'insertion. Cette liste de marqueurs est formée de 92 points d'insertion.

### Requête d'obtention des matrices de position/orientation des vertèbres

Cette requête est exprimée via l'étiquette suivante :

```
GET_ANAT_POSES
```



La réponse à cette requête utilise la structure suivante :

```
ANAT_POSES: v0a0,0 v0a0,1 v0a0,2 v0a0,3 v0a1,0 v0a1,1 v0a1,2 v0a1,3 v0a2,0
v0a2,1 v0a2,2 v0a2,3 v0a3,0 v0a3,1 v0a3,2 v0a3,3 v1a0,0 v1a0,1 v1a0,2 v1a0,3
v1a1,0 v1a1,1 v1a1,2 v1a1,3 v1a2,0 v1a2,1 v1a2,2 v1a2,3 v1a3,0 v1a3,1 v1a3,2
v1a3,3 ... v16a0,0 v16a0,1 v16a0,2 v16a0,3 v16a1,0 v16a1,1 v16a1,2 v16a1,3
v16a2,0 v16a2,1 v16a2,2 v16a2,3 v16a3,0 v16a3,1 v16a3,2 v16a3,3
```

où chaque  $v_k a_{i,j}$  correspond à l'élément de la  $i^e$  colonne et de la  $j^e$  ligne de la matrice de position/orientation de la  $k^e$  vertèbre. Les matrices des 17 vertèbres (de T1 à L5), formant la colonne simulée par le moteur biomécanique, sont transmises via ce type de message.

### Requête d'insertion d'un implant et d'obtention des données d'insertion

Ce type de requête utilise la structure suivante :

```
ADD_IMP Vert:num_vert Cat:cat_imp Mod:mod_imp Pos:px py pz
```

où **num\_vert** correspond au numéro de la vertèbre formé de deux caractères (de "01" à "17"), **cat\_imp** est la catégorie de l'implant ("stSc" pour une vis fixe, "maSc" pour une vis multiaxiale ou "hook" pour un crochet), **mod\_imp** est le modèle de l'implant et **p<sub>x</sub> p<sub>y</sub> p<sub>z</sub>** correspondent aux coordonnées x, y, z de la dernière position de l'implant.

La réponse à ce type de requête utilise la structure suivante :

```
IMP_MAT Vert:num_vert Cat:cat_imp Mod:mod_imp Id:id_imp
Loc:cot_imp Mat:a0,0 a0,1 a0,2 a0,3 a1,0 a1,1 a1,2 a1,3 a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3
```

où **num\_vert** correspond au numéro de la vertèbre formé de deux caractères (de "01" à "17"), **cat\_imp** est la catégorie de l'implant ("stSc" pour une vis fixe, "maSc" pour une vis multiaxiale ou "hook" pour un crochet), **mod\_imp** est le modèle de l'implant, **id\_imp** est le numéro d'identification de l'implant, **cot\_imp** est le côté de la vertèbre

où l'implant est inséré ("L" ou "R") et  $a_{i,j}$  est l'élément de la  $i^e$  colonne et de la  $j^e$  ligne de la nouvelle matrice de position/orientation de l'implant.

### Requête de modification de position d'un implant vertébral et d'obtention des nouvelles données positionnelles

Ce type de requête utilise la structure suivante :

```
SWI_IMP Vert:num_vert Loc:cot_imp
```

où **num\_vert** correspond au numéro de la vertèbre formé de deux caractères (de "01" à "17") et **cot\_imp** est le côté de la vertèbre où l'implant est inséré ("L" ou "R").

La réponse à une requête réussie utilise la structure suivante :

```
SWI_MAT Vert:num_vert Loc:cot_imp Id:id_imp Mat:a0,0 a0,1 a0,2 a0,3
a1,0 a1,1 a1,2 a1,3 a2,0 a2,1 a2,2 a2,3 a3,0 a3,1 a3,2 a3,3
```

où **num\_vert** correspond au numéro de la vertèbre formé de deux caractères (de "01" à "17"), **cot\_imp** est le côté de la vertèbre où l'implant est inséré ("L" ou "R"), **id\_imp** est le numéro d'identification de l'implant et  $a_{i,j}$  est l'élément de la  $i^e$  colonne et de la  $j^e$  ligne de la nouvelle matrice de position/orientation de l'implant.

La réponse une requête qui a échoué utilise la structure suivante :

```
NO_SWITCH_POSSIBLE
```

### Requête de suppression d'un implant vertébral et d'obtention d'une confirmation

Ce type de requête utilise la structure suivante :

```
REM_IMP Vert:num_vert Loc:cot_imp
```

La réponse à ce type de requête utilise la structure suivante :

```
REM_REP Vert:num_vert Loc:cot_imp
```

où **num\_vert** correspond au numéro de la vertèbre formé de deux caractères (de "01" à "17") et **cot\_imp** est le côté de la vertèbre où l'implant est inséré ("L" ou "R").

### Requête d'attachement complète d'une tige

Ce type de requête utilise la structure suivante :

```
NEW_ATA Sty:att_sty Loc:cot_att Pts:p0,x p0,y p0,z p1,x p1,y p1,z p2,x  
p2,y p2,z p3,x p3,y p3,z p4,x p4,y p4,z p5,x p5,y p5,z p6,x p6,y p6,z
```

où **att\_sty** indique l'ordre d'attachement ("TBF" pour un attachement de haut en bas et "BTF" pour un attachement de bas en haut), **cot\_att** est le côté de la colonne où la tige est attachée ("L" ou "R") et **p<sub>i,x</sub> p<sub>i,y</sub> p<sub>i,z</sub>** correspondent aux coordonnées x, y, z du i<sup>e</sup> point de contrôle de la tige. Les tiges sont formées de sept points de contrôle.

Une requête réussie implique l'envoi d'une réponse qui suit la structure présentée dans la sous-section *Structure des réponses aux manœuvres correctrices*. Toutes les requêtes qui provoquent une modification de l'état biomécanique de la colonne, via l'application de forces correctrices, retournent des réponses basées sur cette même structure.

La réponse une requête qui a échoué utilise la structure suivante :

```
NEW_ATA_FAILED
```

### Requête de détachement complet d'une tige

Ce type de requête utilise la structure suivante :

```
ROD_DTA Id:id_tig
```

où **id\_tig** est le numéro d'identification de la tige à détacher.

La réponse une requête qui a échoué utilise la structure suivante :

```
NEW_DTA_FAILED
```

### Requête de rotation de tige

Ce type de requête utilise la structure suivante :

```
ROD_ROT Id:id_tig Ang:tig_rot
```

où **id\_tig** est le numéro d'identification de la tige à détacher et **tig\_rot** est l'angle de rotation désiré (en degrés).

La réponse une requête qui a échoué utilise la structure suivante :

```
ROD_ROT_FAILED
```

### Structure des réponses aux manœuvres correctrices

Toutes les réponses aux requêtes de manœuvres correctrices (attachement et détachement de tige, rotation de tige) utilisent la structure suivante :

```
SPINE_UPDATE_DATA MARKERS: m0,x m0,y m0,z m1,x m1,y m1,z ... m92,x m92,y
m92,z ANAT_POSES: v0a0,0 v0a0,1 v0a0,2 v0a0,3 v0a1,0 v0a1,1 v0a1,2 v0a1,3
v0a2,0 v0a2,1 v0a2,2 v0a2,3 v0a3,0 v0a3,1 v0a3,2 v0a3,3 v1a0,0 v1a0,1 v1a0,2
v1a0,3 v1a1,0 v1a1,1 v1a1,2 v1a1,3 v1a2,0 v1a2,1 v1a2,2 v1a2,3 v1a3,0 v1a3,1
v1a3,2 v1a3,3 ... v16a0,0 v16a0,1 v16a0,2 v16a0,3 v16a1,0 v16a1,1 v16a1,2
v16a1,3 v16a2,0 v16a2,1 v16a2,2 v16a2,3 v16a3,0 v16a3,1 v16a3,2 v16a3,3
RODS_PTS: Id0:id_tig Loc0:cot_att Pts0:p0,x p0,y p0,z p1,x p1,y p1,z
p2,x p2,y p2,z p3,x p3,y p3,z p4,x p4,y p4,z p5,x p5,y p5,z p6,x p6,y p6,z
Id1:id_tig Loc1:cot_att Pts1:p0,x p0,y p0,z p1,x p1,y p1,z p2,x p2,y p2,z
p3,x p3,y p3,z p4,x p4,y p4,z p5,x p5,y p5,z p6,x p6,y p6,z ... Idn:id_tig
Locn:cot_att Ptsn:p0,x p0,y p0,z p1,x p1,y p1,z p2,x p2,y p2,z p3,x p3,y p3,z
p4,x p4,y p4,z p5,x p5,y p5,z p6,x p6,y p6,z IMPS_MAT Id0:id_imp Mat0:a0,0
a0,1 a0,2 a0,3 a1,0 a1,1 a1,2 a1,3 a2,0 a2,1 a2,2 a2,3 a3,0 a3,1 a3,2 a3,3
Id1:id_imp Mat1:a0,0 a0,1 a0,2 a0,3 a1,0 a1,1 a1,2 a1,3 a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3 ... Idn:id_imp Matn:a0,0 a0,1 a0,2 a0,3 a1,0 a1,1 a1,2
a1,3 a2,0 a2,1 a2,2 a2,3 a3,0 a3,1 a3,2 a3,3
```

où chaque trio  $m_{i,x}$   $m_{i,y}$   $m_{i,z}$  correspond aux coordonnées x, y, z du i<sup>e</sup> point d'insertion,  $v_k a_{i,j}$  correspond à l'élément de la i<sup>e</sup> colonne et de la j<sup>e</sup> ligne de la matrice de position/orientation de la k<sup>e</sup> vertèbre,  $Id_i$  (lorsqu'il est entre l'étiquette **RODS\_PTS** et la suivante) indique que l'information qui suit concernent la i<sup>e</sup> tige, **id\_tig** est le numéro d'identification de la tige, **Loc<sub>i</sub>** indique que le côté d'attachement de la i<sup>e</sup> tige va suivre, **cot\_att** est le côté de la colonne où la tige est attachée, **Pts<sub>i</sub>** indique que les points de contrôle de la i<sup>e</sup> tiges vont suivre,  $p_{i,x}$   $p_{i,y}$   $p_{i,z}$  correspondent aux nouvelles coordonnées x, y, z du i<sup>e</sup> point de contrôle de la tige,  $Id_i$  (lorsqu'il est après l'étiquette **IMPS\_MAT**) indique que l'information qui suit concernent le i<sup>e</sup> implant, **id\_imp** est le numéro d'identification de l'implant, **Mat<sub>i</sub>** indique que la matrice du i<sup>e</sup> implant va suivre et  $a_{i,j}$  est l'élément de la i<sup>e</sup> colonne et de la j<sup>e</sup> ligne de la nouvelle matrice de position/orientation de l'implant.

## ANNEXE IV : Exemples de messages réseaux envoyés ou reçus par le serveur biomécanique

Cette annexe présente des exemples de messages réseaux pouvant être transmis entre les clients et le serveur biomécanique lors de requêtes d'instrumentation du rachis scoliotique d'un patient spécifique.

### Requête d'obtention des marqueurs des points d'insertion pré-calculés

```
MARKERS: -370.0327 9.8685 252.0420 -377.0613 7.6180 234.6061 -378.9668 5.0796 212.0726
-380.5687 -2.9045 189.8577 -380.5618 -9.9707 171.2357 -377.8467 -22.4701 150.4321 -
378.5714 -29.0547 133.0117 -374.7614 -31.4219 112.8832 -372.2814 -30.3746 92.0861 -
369.7150 -22.1579 67.5798 -361.4329 -14.6034 41.2315 -352.9031 -3.4016 8.9487 -
344.9278 9.4681 -21.5657 -335.6855 19.7030 -49.0265 -329.1455 25.9041 -83.8244 -
330.7100 29.0267 -114.0513 -339.8040 29.2401 -143.6501 -369.3834 -20.0193 250.1050 -
378.2501 -15.4189 234.1926 -378.0330 -12.9785 214.7710 -378.2271 -18.0958 197.2774 -
382.6158 -27.1593 180.9184 -380.3726 -38.4230 160.6658 -380.0100 -46.4212 140.6011 -
378.5649 -48.4357 115.1263 -377.1536 -47.4441 89.0743 -372.1118 -40.6872 61.3487 -
364.2265 -36.7234 32.7341 -354.7073 -25.2434 2.4930 -346.7260 -15.5466 -28.2094 -
337.1431 -9.4402 -54.4605 -330.6407 -2.4013 -86.1415 -332.7504 -2.8573 -113.9964 -
339.9385 -9.0690 -141.9597 -366.3021 8.7384 254.3066 -372.1152 8.1529 237.0386 -
375.6041 5.6348 213.9246 -377.9982 -1.6833 192.0033 -377.0587 -8.6978 175.1528 -
374.7079 -19.8075 155.0501 -374.4158 -28.8061 137.9190 -371.5250 -32.0157 117.9743 -
370.9498 -30.1550 97.5100 -367.4989 -25.8983 71.7279 -360.2765 -16.5446 47.4114 -
352.5471 -6.7231 15.0864 -367.3535 -19.7550 251.7175 -371.5754 -15.4312 237.0016 -
374.2841 -13.2144 217.6382 -375.5707 -17.7381 200.1040 -379.2516 -25.1348 185.3266 -
377.4234 -35.9378 165.4022 -377.1608 -45.4125 144.8432 -376.1902 -49.3661 119.9865 -
375.3534 -48.3470 93.9503 -370.8736 -42.4599 66.1275 -363.2086 -39.9898 38.4657 -
354.7552 -26.2049 8.5727 -377.4962 2.9102 238.4017 -379.3942 2.3413 222.3767 -381.3641
-1.1843 199.5398 -381.6395 -10.3272 177.9678 -379.9546 -20.3580 159.5765 -377.9419 -
33.7269 139.2259 -377.3299 -37.5390 119.7201 -372.1382 -35.0563 98.0939 -368.7682 -
31.9724 74.6725 -364.4417 -21.7194 50.4142 -356.4516 -13.8156 19.5688 -349.0742 -
2.7717 -11.4847 -340.2737 8.4716 -43.6063 -335.0441 17.5259 -72.6111 -332.6726 20.3954
-105.1128 -341.8903 17.4937 -132.6812 -355.8616 17.9654 -151.4820 -376.4556 -11.8435
239.4901 -380.1075 -11.5947 221.9508 -380.2263 -13.6091 201.5025 -379.6513 -20.7560
183.0096 -381.4844 -31.1626 166.2253 -379.4369 -43.4604 145.2756 -379.1940 -48.1288
123.9895 -375.3343 -46.3357 99.8754 -370.9291 -42.4772 73.0402 -366.6917 -31.5677
46.5486 -357.2440 -21.4556 16.3207 -350.6386 -12.1878 -14.2078 -341.4633 -1.1188 -
47.0949 -336.1422 6.3143 -74.3456 -334.4322 8.2106 -106.4501 -342.9434 3.6586 -
133.2709 -355.2364 1.0036 -150.6216
```

### Requête d'obtention des matrices de position/orientation des vertèbres

```
ANAT_POSES: 0.9972 -0.0000 0.0742 -348.7335 0.0065 0.9961 -0.0876 -6.6548 -0.0739
0.0878 0.9934 244.8082 0.9931 -0.0265 0.1141 -350.5191 0.0295 0.9993 -0.0243 -5.2304 -
0.1134 0.0275 0.9932 225.2198 0.9768 -0.1639 0.1380 -353.1548 0.1176 0.9485 0.2943 -
7.8315 -0.1791 -0.2712 0.9457 207.0813 0.9877 -0.1454 0.0576 -355.5127 0.1084 0.9020
0.4180 -0.4127 -0.1127 -0.4066 0.9066 188.2033 0.9938 0.1111 0.0016 -356.1097 -0.0955
0.8467 0.5234 -25.9294 0.0568 -0.5204 0.8521 170.3802 0.9906 0.1366 0.0056 -354.6488 -
0.1264 0.8999 0.4173 -36.5380 0.0520 -0.4141 0.9088 152.4802 0.9841 0.1697 -0.0528 -
353.6155 -0.1463 0.9422 0.3014 -45.0258 0.1009 -0.2889 0.9520 131.9043 0.9716 0.2144 -
0.0999 -351.6601 -0.2129 0.9767 0.0257 -48.8889 0.1031 -0.0036 0.9947 110.6606 0.9547
0.2252 -0.1945 -348.0531 -0.2488 0.9627 -0.1063 -46.6127 0.1633 0.1499 0.9751 87.6272
```

```

0.9621 0.1759 -0.2086 -342.0047 -0.2442 0.8959 -0.3712 -39.2216 0.1216 0.4081 0.9048
63.1202 0.9509 0.0785 -0.2994 -334.3079 -0.1950 0.9031 -0.3825 -28.1362 0.2403 0.4221
0.8741 37.5073 0.9612 0.0608 -0.2691 -325.3158 -0.1512 0.9319 -0.3296 -15.8463 0.2308
0.3575 0.9050 10.8435 0.9456 0.0430 -0.3224 -314.6227 -0.1372 0.9515 -0.2753 -7.8816
0.2950 0.3045 0.9057 -18.8243 0.9749 0.0143 -0.2221 -303.2716 -0.0798 0.9540 -0.2891
3.2584 0.2077 0.2996 0.9312 -51.5043 0.9959 0.0551 -0.0722 -295.0680 -0.0617 0.9938 -
0.0928 10.6647 0.0666 0.0969 0.9931 -86.9751 0.9825 0.0603 0.1762 -295.8291 -0.0761
0.9936 0.0840 11.1712 -0.1700 -0.0959 0.9808 -123.7939 0.8876 -0.0071 0.4606 -308.4718
-0.0578 0.9903 0.1267 7.0717 -0.4570 -0.1391 0.8785 -157.6155

```

### Requête d'insertion d'un implant et d'obtention des données d'insertion

```
ADD_IMP Vert:11 Cat:hook Mod:ImplMod858102 Pos:0.8200 2.6568 -0.0456
```

```
IMP_MAT Vert:11 Cat:hook Mod:ImplMod858102 Id:120231 Loc:L Mat:-0.1356 -0.6383 -0.7578
-378.6989 -0.3989 -0.6650 0.6315 -33.1706 -0.9069 0.3879 -0.1644 130.6146
```

### Requête de modification de position d'un implant vertébral et d'obtention des nouvelles données positionnelles

```
SWI_IMP Vert:10 Loc:L
```

```
SWI_MAT Vert:10 Loc:L Id:120731 Mat:0.2493 0.6745 -0.6949 -365.6274 0.3211 -0.7345 -
0.5978 -26.3609 -0.9136 -0.0741 -0.3997 68.6635
```

### Requête de suppression d'un implant vertébral et d'obtention d'une confirmation

```
REM_IMP Vert:09 Loc:R
```

```
REM_REP Vert:09 Loc:R
```

### Requête d'attachement complète d'une tige

```
NEW_ATA Sty:TBF Loc:L Pts:-383.1818 -30.9140 207.1748 -383.1818 -30.9140 184.9348 -
383.1818 -30.9140 162.6948 -383.1818 -30.9140 140.4548 -383.1818 -30.9140 118.2148 -
383.1818 -30.9140 95.9748 -383.1818 -30.9 140 73.7348
```

```

SPINE_UPDATE_DATA MARKERS: -370.0327 9.8685 252.0420 -377.0613 7.6180 234.6061 -
378.9668 5.0796 212.0726 -380.5687 -2.9045 189.8577 -380.5618 -9.9707 171.2357 -
377.8467 -22.4701 150.4321 -378.5714 -29.0547 133.0117 -374.7614 -31.4219 112.8832 -
372.2814 -30.3746 92.0861 -369.7150 -22.1579 67.5798 -361.4329 -14.6034 41.2315 -
352.9031 -3.4016 8.9487 -344.9278 9.4681 -21.5657 -335.6855 19.7030 -49.0265 -329.1455
25.9041 -83.8244 -330.7100 29.0267 -114.0513 -339.8040 29.2401 -143.6501 -369.3834 -
20.0193 250.1050 -378.2501 -15.4189 234.1926 -378.0330 -12.9785 214.7710 -378.2271 -
18.0958 197.2774 -382.6158 -27.1593 180.9184 -380.3726 -38.4230 160.6658 -380.0100 -
46.4212 140.6011 -378.5649 -48.4357 115.1263 -377.1536 -47.4441 89.0743 -372.1118 -
40.6872 61.3487 -364.2265 -36.7234 32.7341 -354.7073 -25.2434 2.4930 -346.7260 -
15.5466 -28.2094 -337.1431 -9.4402 -54.4605 -330.6407 -2.4013 -86.1415 -332.7504 -
2.8573 -113.9964 -339.9385 -9.0690 -141.9597 -366.3021 8.7384 254.3066 -372.1152
8.1529 237.0386 -375.6041 5.6348 213.9246 -377.9982 -1.6833 192.0033 -377.0587 -8.6978
175.1528 -374.7079 -19.8075 155.0501 -374.4158 -28.8061 137.9190 -371.5250 -32.0157
117.9743 -370.9498 -30.1550 97.5100 -367.4989 -25.8983 71.7279 -360.2765 -16.5446
47.4114 -352.5471 -6.7231 15.0864 -367.3535 -19.7550 251.7175 -371.5754 -15.4312
237.0016 -374.2841 -13.2144 217.6382 -375.5707 -17.7381 200.1040 -379.2516 -25.1348
185.3266 -377.4234 -35.9378 165.4022 -377.1608 -45.4125 144.8432 -376.1902 -49.3661
119.9865 -375.3534 -48.3470 93.9503 -370.8736 -42.4599 66.1275 -363.2086 -39.9898
38.4657 -354.7552 -26.2049 8.5727 -377.4962 2.9102 238.4017 -379.3942 2.3413 222.3767
-381.3641 -1.1843 199.5398 -381.6395 -10.3272 177.9678 -379.9546 -20.3580 159.5765 -
377.9419 -33.7269 139.2259 -377.3299 -37.5390 119.7201 -372.1382 -35.0563 98.0939 -
368.7682 -31.9724 74.6725 -364.4417 -21.7194 50.4142 -356.4516 -13.8156 19.5688 -
349.0742 -2.7717 -11.4847 -340.2737 8.4716 -43.6063 -335.0441 17.5259 -72.6111 -
332.6726 20.3954 -105.1128 -341.8903 17.4937 -132.6812 -355.8616 17.9654 -151.4820 -
376.4556 -11.8435 239.4901 -380.1075 -11.5947 221.9508 -380.2263 -13.6091 201.5025 -
379.6513 -20.7560 183.0096 -381.4844 -31.1626 166.2253 -379.4369 -43.4604 145.2756 -
379.1940 -48.1288 123.9895 -375.3343 -46.3357 99.8754 -370.9291 -42.4772 73.0402 -
366.6917 -31.5677 46.5486 -357.2440 -21.4556 16.3207 -350.6386 -12.1878 -14.2078 -
341.4633 -1.1188 -47.0949 -336.1422 6.3143 -74.3456 -334.4322 8.2106 -106.4501 -
342.9434 3.6586 -133.2709 -355.2364 1.0036 -150.6216 ANAT_POSES: 0.9972 -0.0000 0.0742
-348.7335 0.0065 0.9961 -0.0876 -6.6548 -0.0739 0.0878 0.9934 244.8082 0.9931 -0.0265
0.1141 -350.5191 0.0295 0.9993 -0.0243 -5.2304 -0.1134 0.0275 0.9932 225.2198 0.9768 -
0.1639 0.1380 -353.1548 0.1176 0.9485 0.2943 -7.8315 -0.1791 -0.2712 0.9457 207.0813
0.9877 -0.1454 0.0576 -355.5127 0.1084 0.9020 0.4180 -14.3242 -0.1127 -0.4066 0.9066
188.2033 0.9938 0.1111 0.0016 -356.1097 -0.0955 0.8467 0.5234 -25.9294 0.0568 -0.5204
0.8521 170.3802 0.9906 0.1366 0.0056 -354.6488 -0.1264 0.8999 0.4173 -36.5380 0.0520 -
0.4141 0.9088 152.4802 0.9841 0.1697 -0.0528 -353.6155 -0.1463 0.9422 0.3014 -45.0258
0.1009 -0.2889 0.9520 131.9043 0.9716 0.2144 -0.0999 -351.6601 -0.2129 0.9767 0.0257 -
48.8889 0.1031 -0.0036 0.9947 110.6606 0.9547 0.2252 -0.1945 -348.0531 -0.2488 0.9627
-0.1063 -46.6127 0.1633 0.1499 0.9751 87.6272 0.9621 0.1759 -0.2086 -342.0047 -0.2442
0.8959 -0.3712 -39.2216 0.1216 0.4081 0.9048 63.1202 0.9509 0.0785 -0.2994 -334.3079 -
0.1950 0.9031 -0.3825 -28.1362 0.2403 0.4221 0.8741 37.5073 0.9612 0.0608 -0.2691 -
325.3158 -0.1512 0.9319 -0.3296 -15.8463 0.2308 0.3575 0.9050 10.8435 0.9456 0.0430 -
0.3224 -314.6227 -0.1372 0.9515 -0.2753 -7.8816 0.2950 0.3045 0.9057 -18.8243 0.9749
0.0143 -0.2221 -303.2716 -0.0798 0.9540 -0.2891 3.2584 0.2077 0.2996 0.9312 -51.5043
0.9959 0.0551 -0.0722 -295.0680 -0.0617 0.9938 -0.0928 10.6647 0.0666 0.0969 0.9931 -
86.9751 0.9825 0.0603 0.1762 -295.8291 -0.0761 0.9936 0.0840 11.1712 -0.1700 -0.0959
0.9808 -123.7939 0.8876 -0.0071 0.4606 -308.4718 -0.0578 0.9903 0.1267 7.0717 -0.4570
-0.1391 0.8785 -157.6155 RODS_PTS: Id:1 Loc:L Pts:-384.5006 -32.1563 169.6040 -
381.9928 -32.2103 147.5059 -379.4818 -32.2690 125.4078 -377.1611 -32.1265 103.2883 -
375.1079 -31.6732 81.1472 -373.0580 -31.2152 59.0065 -371.0081 -30.7572 36.8659
IMPS_MAT: Id:120731 Mat:0.1128 -0.6085 -0.7855 -380.7390 -0.0024 -0.7907 0.6122 -
32.2373 -0.9936 -0.0671 -0.0906 136.4569 Id:120931 Mat:0.0922 -0.7471 -0.6583 -
376.0613 0.0206 -0.6596 0.7514 -31.8862 -0.9955 -0.0828 -0.0454 91.4450 Id:121041
Mat:0.1830 0.3628 -0.9137 -374.8536 0.3167 -0.9016 -0.2946 -39.5829 -0.9307 -0.2355 -
0.2799 64.5906

```



**Requête de détachement complet d'une tige**

ROD\_DTA Id:1

**Requête de rotation de tige**

ROD\_ROT Id:1 Ang:5

## ANNEXE V : Diagramme de classe du serveur de téléprésence et explications

La figure V.1 présente le diagramme de classe simplifié du serveur de téléprésence. Son design inhérent est ainsi dévoilé afin d'en permettre une meilleure compréhension. L'ensemble des classes de la boîte C++ Sockets Library correspond au module *Interface réseau* de la figure 5.1 du chapitre 5. La classe **TelepresenceServer** correspond au module de gestion de téléprésence.

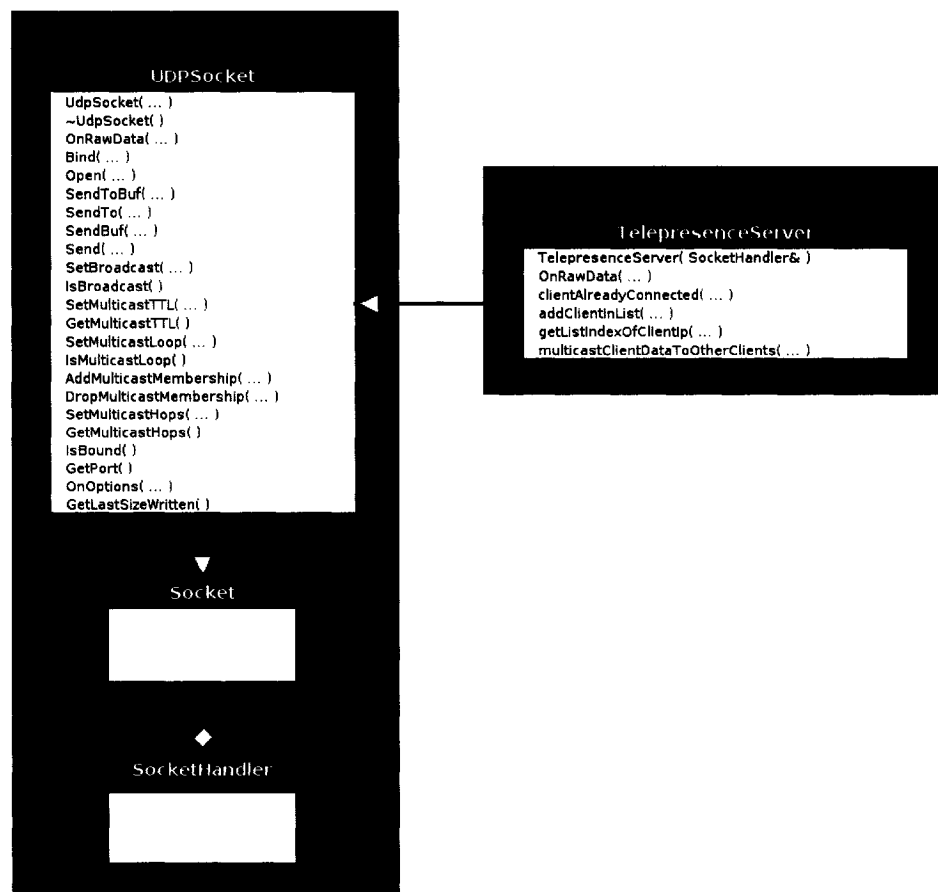


Figure V.1 : Composantes du serveur de téléprésence multi-usagers.

La classe **UDPSocket** provient de la librairie C++ Sockets Library et implémente le protocole UDP. Elle dérive de la super-classe **Socket** dont laquelle dérivent d'autres

classes implémentant divers protocoles réseaux. La classe **SocketHandler** joue le rôle de contrôleur, de gestionnaire d'événements et de répartiteur pour les classes qui dérivent de **Socket**. Elle gère les événements (lecture/écriture/exception) de la classe **TelepresenceServer** et les lui reporte à l'aide de méthodes de « callback ».

La fonction de « callback » **OnRawData()** est appelée lors de la réception de données brutes exprimées sous la forme de chaînes de caractères. **OnRawData()** vérifie d'abord, à l'aide de la fonction **clientAlreadyConnected()**, si les données proviennent d'un client connu par le serveur. Un client inconnu est identifié par son adresse IP et ajouté à la liste des participants via la fonction **addClientInList()**. Un numéro d'index est également généré afin de pouvoir l'identifier avec plus de facilité. La fonction **getListIndexOfClientIp()** permet d'ailleurs d'obtenir le numéro d'index associé à l'adresse IP d'un client préalablement identifié. **OnRawData()** transmet finalement les données de positionnelles de la tête et de la souris 3D du client (qu'il soit nouvellement identifié ou non) à tous les autres clients via la fonction **multicastClientDataToOtherClients()**.

## ANNEXE VI : Structures des messages réseaux envoyés ou reçus par le serveur de téléprésence

Les messages de positions de tête envoyés au serveur utilisent la forme suivante :

```
HEAD: a0,0 a0,1 a0,2 a0,3 a1,0 a1,1 a1,2 a1,3 a2,0 a2,1 a2,2 a2,3 a3,0 a3,1
a3,2 a3,3
```

où **HEAD** est une étiquette qui permet au serveur de reconnaître le type de message et  $a_{i,j}$  est l'élément de la  $i^{\text{e}}$  colonne et de la  $j^{\text{e}}$  ligne de la nouvelle matrice de position/orientation de la tête du client.

Les messages de positions de la souris 3D utilisent la même structure. La seule distinction est l'étiquette :

```
WAND: a0,0 a0,1 a0,2 a0,3 a1,0 a1,1 a1,2 a1,3 a2,0 a2,1 a2,2 a2,3 a3,0 a3,1
a3,2 a3,3
```

Les messages de positions de tête propagés du serveur aux clients utilisent la structure suivante :

```
From:id_snd To:id_rec HEAD: a0,0 a0,1 a0,2 a0,3 a1,0 a1,1 a1,2 a1,3 a2,0
a2,1 a2,2 a2,3 a3,0 a3,1 a3,2 a3,3
```

où **id\_snd** correspond au numéro d'identification du client qui envoie sa position, **id\_rec** est le numéro d'identification du client à qui est destiné le message et  $a_{i,j}$  est l'élément de la  $i^{\text{e}}$  colonne et de la  $j^{\text{e}}$  ligne de la nouvelle matrice de position/orientation de la tête du client.

Les messages de positions de la souris 3D, propagés aux clients, utilisent la même structure. La seule distinction est que l'étiquette **HEAD** est remplacée par **WAND** :

```
From:id_snd To:id_rec WAND: a0,0 a0,1 a0,2 a0,3 a1,0 a1,1 a1,2 a1,3 a2,0
a2,1 a2,2 a2,3 a3,0 a3,1 a3,2 a3,3
```

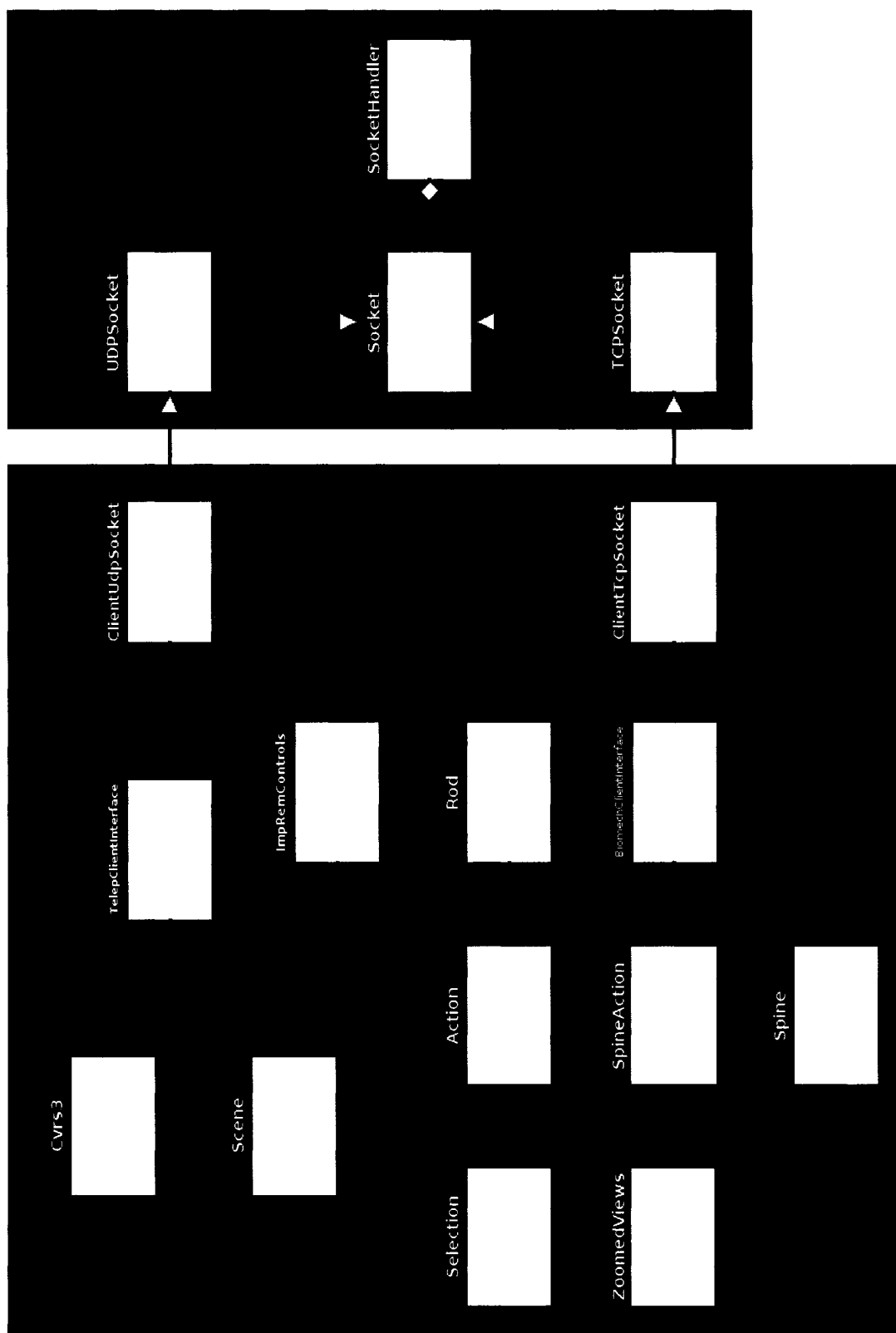
## ANNEXE VII : Diagramme de classe du client de simulation et explications

La figure VII.1 présente le diagramme de classe simplifié du client. Les noms des fonctions ont été omis afin d'assurer une meilleure lisibilité.

La classe **Cvrs3** est la classe maîtresse du simulateur. Elle s'occupe du démarrage et de la gestion globale de la simulation. Les fonctions d'initialisation suivantes, membres de l'API de VR Juggler, sont implémentées : **init()** et **initScene()**. Celles-ci sont appelées avant que toute tâche d'affichage ne soit effectuée. La fonction **init()** s'occupe de :

- Créer la structure de données qui sérialise certaines données (réponses des serveurs) au travers le réseau afin qu'elles soient disponibles pour les machines faisant partie de la grappe de la CAVE (en mode immersif) ;
- Initialiser le gestionnaire de graphe de scène OpenSG ;
- Créer une instance de la classe Scene ;
- Initier les communications avec le serveur biomécanique et le serveur de téléprésence.

La fonction **initScene()** s'occupe de l'initialisation des variables qui pointent vers la matrice de position de la souris 3D, l'état de ses trois boutons (pressés ou non) et la matrice de position des lunettes. Elle s'occupe également de générer le graphe de scène initial afin que la scène soit prête à être affichée.



**Figure VII.1 :** Diagramme de classe du client de simulation de chirurgie scoliotique en réalité

La classe **Cvrs3** implémente également des fonctions faisant partie de la boucle séquentielle de VR Juggler qui permettent d'effectuer des traitements avant, pendant et après l'affichage d'une image : **preFrame()**, **draw()** et **postFrame()**. Celles-ci sont appelées séquentiellement dans une boucle qui ne termine que lorsque explicitement demandé. La fonction **preFrame()** est appelée, comme son nom l'indique, juste avant l'affichage d'une image. Elle est utilisée pour mettre à jour les variables d'état de la souris 3D et des lunettes afin que la scène puisse être adaptée le plus rapidement possible en fonction de la perspective de l'utilisateur. La scène est subséquemment affichée par l'appel de **draw()**. Celle-ci ne fait qu'appeler une fonction OpenSG du même nom qui gère implicitement l'affichage en fonction de l'état du graphe de scène. La gestion du graphe de scène en fonction des actions de l'utilisateur, exprimées via la souris 3D, est effectuée dans **postFrame()**. La fonction **manageInteraction(...)**, membre de la classe **Scene**, est appelée à cette fin. Effectuer cette mise à jour juste après l'affichage de la dernière image pourrait, à première vue, sembler être une mauvaise décision. Ce choix a été fait puisque la complexité des traitements effectués dans **manageInteraction(...)** nuit au taux de rafraîchissement global de la simulation lorsqu'ils sont effectués dans **preFrame()**. Cette fonction de VR Juggler est conçue dans l'optique d'une utilisation limitée aux traitements simples de dernière minute.

La classe **Scene** s'occupe principalement de la gestion du graphe de scène. Elle est un point d'entrée vers ce graphe et permet de créer, de supprimer et de modifier des nœuds<sup>18</sup>. La fonction **initAllScene()** est appelée lors du démarrage pour créer les nœuds qui sont nécessaires à l'état initial de la scène: lumières, transformations et objets géométriques. Les modèles géométriques utilisés (la plupart étant des fichiers VRML) sont chargés via la fonction **loadModels()**. La plupart des objets sont initialement intégrés dans le graphe et sont combinés à des nœuds de transformation afin d'être

---

<sup>18</sup> Consulter la section 3.2.2 pour une revue du concept de nœud OpenSG.

visibles et adéquatement disposés au démarrage de l'application. Les modèles restants (la majorité des implants par exemple) sont conservés en mémoire dans l'attente qu'une action de l'utilisateur nécessite leur intégration. La fonction **manageInteraction(...)** s'occupe notamment, via la classe **TelepClientInterface**, de transmettre les nouvelles positions de l'utilisateur au serveur de téléprésence et de mettre à jours la position des avatars des participants. Elle gère les événements associés à la souris 3D et, à l'aide de la classe **Selection**, détermine si un objet peut être sélectionné. Elle gère aussi, via la classe **Action**, les actions effectuées sur les objets de la scène qui permettent une interaction.

La classe **Selection** maintient une liste des objets pouvant être sélectionnés. Les fonctions **addSelectableObj(...)** et **remSelectableObj(...)** permettent d'ajouter et de supprimer un objet dans cette liste. La fonction **objNearEnough(...)** détermine si la souris 3D est assez près d'un ou de plusieurs objets pour permettre sa sélection. La fonction **getSelectedObjIndex()** retourne ensuite le nœud de l'objet le plus près (ou du seul objet étant assez près). La souris 3D doit être inactive pour permettre une sélection. Un objet ne peut, par exemple, être sélectionné si elle est déjà utilisée pour déplacer un implant.

La classe **Scene** offre également des fonctions qui permettent de verrouiller, déverrouiller la sélection d'un objet déjà sélectionné et de forcer la sélection d'un objet non sélectionné.

La classe **Action** gère le déplacement (« dragging ») des implants/outils et les actions effectuées sur les manettes, la lampe et la colonne. Sa relation avec la classe **Selection** lui permet de déterminer si une action est possible puisque toute action nécessite la sélection préalable d'un objet. La plupart des actions et des conséquences à celles-ci impliquent une mise à jour du graphe de scène qui est directement effectuée par la classe **Scene**. Certaines actions plus complexes sont, cependant, effectuées via des classes dédiées qui gèrent la colonne vertébrale, les tiges métalliques, les vues



alternatives et les manettes de contrôle de sélection d'implant : **SpineAction**, **Spine**, **Rod**, **ZoomedViews** et **ImpRemControls**.

La classe **SpineAction** offre des fonctions qui permettent d'insérer un implant dans une vertèbre, modifier la position d'un implant, enlever un implant, attacher et détacher une tige et effectuer une rotation de tige. Elle interagit avec la classe **ClientTCPSocket** afin de formuler et d'envoyer au serveur biomécanique les requêtes associées à ces actions. Elle utilise finalement les réponses obtenues afin de mettre à jour, via la classe **Spine**, l'état géométrique de la colonne vertébrale. Les fonctions d'instrumentation qu'elle propose sont implémentées à l'aide de « threads ». L'objectif est d'assurer une utilisation continue du simulateur malgré le fait qu'il attend une réponse du serveur.

L'état géométrique de la colonne vertébrale est géré et conservé par la classe **Spine**. Elle gère et conserve, plus précisément, les matrices de transformation de chaque vertèbre, les marqueurs d'insertion pré-calculés, l'état d'instrumentation de la colonne, les numéros d'identification des tiges et des implants attachés.

La classe **Rod** gère l'état géométrique d'une tige spécifique. Elle permet une modification des matrices de transformation des points de contrôle et des segments qui la représente. La relation qu'elle entretient avec la classe **Action** permet à l'utilisateur de la déplacer ou de modifier la position de ses points de contrôle. L'utilisateur peut ainsi lui donner la courbure désirée et la déplacer à proximité de la colonne afin qu'elle soit attachée aux implants à proximité.

La classe **ZoomedView** gère l'interaction avec la manette de contrôle de la transparence et du zoom des vues alternatives. Ces vues peuvent être utilisées pour afficher un implant non-implanté ou une vertèbre implantée. La relation qui existe entre **ZoomedView** et la classe **Selection** permet l'affichage de l'objet qui est ou était préalablement sélectionné. **ZoomedView** conserve en mémoire des pointeurs vers des

copies de l'implant ou de la vertèbre sélectionnée afin d'offrir quatre points de vue paramétrables en terme de transparence et de taille. La modification de ces paramètres (augmenter ou diminuer le zoom, augmenter ou diminuer la transparence) est contrôlée par les quatre boutons de la manette. **ZoomedView** effectue une mise à jour de ces vues lorsqu'un des boutons est appuyé. Il modifie également la couleur d'un bouton (il devient rose) lorsqu'une pression est exercée.

La classe **ImpRemControls** gère l'interaction avec les trois manettes de contrôle de sélection d'implant. Ces manettes permettent de choisir l'implant couramment affiché sur la plateau pour chaque catégorie d'implants : crochets, vis pédiculaires fixes et vis pédiculaires multiaxiales. Les boutons « précédent » et « suivant » permettent de cycliser parmi les douze crochets, six vis pédiculaires fixes et cinq vis pédiculaires multiaxiales disponibles. **ImpRemControls** modifie le nœud courant sélectionné pour une catégorie d'implants lorsqu'une pression est exercée sur un des boutons. Elle gère également l'apparence du bouton pressé : il s'abaisse et change de couleur.

Finalement, les classes **ClientTCPSocket** et **ClientUDPSocket** sont utilisées, respectivement, pour initier la connexion ou l'initiation des communications avec le serveur biomécanique et le serveur de téléprésence. Les relations qu'elles entretiennent avec les classes de C++ Socket Library leurs permettent de gérer l'envoi et la réception de messages. Les relations entretenues avec ces classes sont quasi-identiques à celles décrites dans les sections 4.1.2 et 5.1.2 pour les classes **BiomechServerInterface** et **TelepresenceServer**. Consultez ces sections pour une meilleure compréhension des classes **TCPSocket**, **UDPSocket**, **Socket** et **SocketHandler**.

## ANNEXE VIII : Formulaire d'évaluation

Voici le questionnaire d'évaluation ayant été distribué aux participants des tests usagers.

<p align="center"><b>SIMULATEUR COLLABORATIF DE CHIRURGIE DU RACHIS SCOLIOTIQUE EN RÉALITÉ VIRTUELLE</b></p> <p align="center"><b>Étude d'utilisabilité</b></p> <p align="center"><b>Questionnaire d'évaluation</b></p>
---

Informations personnelles	
Nom :	
Âge :	
Profession :	
Expérience avec jeux vidéos (oui/non) :	
Expérience préalable avec S3 (oui/non) :	
Expérience préalable avec CAVE (oui/non) :	

Veuillez SVP répondre aux prochaines questions par un chiffre de 1 à 5 où :

- 1 signifie « Tout à fait en désaccord » / « Médiocre »
- 2 signifie « Pas vraiment d'accord » / « Faible »
- 3 signifie « Assez d'accord » / « Acceptable »
- 4 signifie « Bien d'accord » / « Bien »
- 5 signifie « Tout à fait d'accord » / « Excellent ».

Inscrivez « X » si la question ne s'applique pas.

<b>Réalisme</b>	
Patient:	/5
Salle et matériel (armoires, table, lampes, éclairage, patient, manettes de contrôle, ...) :	/5
Implants chirurgicaux :	/5
Colonne vertébrale :	/5
Avatars des autres participants (objets les représentant) :	/5
<b>Commentaires :</b>	

<b>Interface immersive (dans la CAVE)</b>	
Facilité d'apprentissage générale :	/5
Facilité d'utilisation générale :	/5
Interface généralement intuitive, prédictible :	/5
Facilité à sélectionner et déplacer des objets :	/5
Facilité d'application de force (formation de la tige, manoeuvre de rotation de tige, ...) :	/5
<b>Commentaires :</b>	

<b>Interface non-immersive (sur poste de travail conventionnel)</b>	
Facilité d'apprentissage générale :	/5
Facilité d'utilisation générale :	/5
Interface généralement intuitive, prédictible :	/5
Facilité à sélectionner et déplacer des objets :	/5
Facilité d'application de force (formation de la tige, manoeuvre de rotation de tige, ...) :	/5
<b>Commentaires :</b>	

<b>Immersion</b>	
Sentiment d'immersion général (dans la CAVE) :	/5
Fluidité et temps de réponse général :	/5
Fluidité et temps de réponse lors d'actions sur la colonne vertébrale :	/5
Fluidité du déplacement des avatars des participants :	/5
<b>Commentaires :</b>	

<b>Tâches</b>	
Planification chirurgicale réaliste :	/5
Réponses réalistes du modèle biomécanique de la colonne :	/5
<b>Commentaires :</b>	

### Collaboration

Perception de la présence des collaborateurs :	/5
Perception des intentions des collaborateurs :	/5
Perception des conséquences des actions des collaborateurs :	/5
Coordination des actions avec les collaborateurs :	/5
Facilité de communication des intentions entre collaborateurs :	/5
<b>Commentaires :</b>	

### Utilité future

Formation générale des médecins résidents :	/5
Planification et entraînement chirurgical cas par cas :	/5
Compréhension de la biomécanique du rachis scoliotique :	/5
Télé-enseignement de l'instrumentation du rachis scoliotique (session collaborative avec chirurgiens experts et médecins résidents):	/5
Télé-entraînement chirurgical avec coordination/collaboration entre chirurgiens :	/5
Télé-planification chirurgicale cas par cas avec coordination/collaboration entre chirurgiens :	
<b>Commentaires :</b>	

**Autres commentaires :**